
# A Stochastic Game Theoretical Model for Cyber Security

THESIS

Michael T. Larkin, Captain, USAF

AFIT-ENS-MS-19-M-133

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

AFIT-ENS-MS-19-M-133

# A STOCHASTIC GAME THEORETICAL MODEL FOR CYBER SECURITY

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Michael T. Larkin, B.S.

Captain, USAF

21 March 2019

AFIT-ENS-MS-19-M-133

# A STOCHASTIC GAME THEORETICAL MODEL FOR CYBER SECURITY

THESIS

Michael T. Larkin, B.S.
Captain, USAF

Committee Membership:

Dr. Darryl K. Ahner, Ph.D.
Chair

Dr. Robert F. Mills, Ph.D.
Reader

# Abstract

The resiliency of systems integrated through cyber networks is of utmost importance due to the reliance on these systems for critical services such as industrial control systems, nuclear production, and military weapons systems. Current research in cyber resiliency remains largely limited to methodologies utilizing a singular technique that is predominantly theoretical with limited examples given. This research uses notional data in presenting a novel approach to cyber system analysis and network resource allocation by leveraging multiple techniques including game theory, stochastic processes, and mathematical programming. An operational network security problem consisting of 20 tactical normal form games provides an assessment of the resiliency of a cyber defender's network by leveraging the solutions of each tactical game to inform transitional probabilities of a discrete-time Markov chain over an attacker-defender state space. Furthermore, the Markov chain provides an assessment of the conditional path through the operational problem with an expected cost of damage to the defender network. The solutions of the tactical games and, in turn the operational problem, are utilized to determine the effects and risks of projected network improvement resource allocation decisions via an integer program. These results can be used to inform network analysts of the resiliency of their network while providing recommendations and requirements for improving their network resiliency posture against potential malicious external actors.

*For my family, friends, and colleagues.*

*Thank you for for making all of this possible.*

# Acknowledgements

I would like to thank my advisor, Dr. Darryl Ahner, for his patience, wisdom, and guidance throughout the writing of this work. I would also like to thank my family, friends, colleagues, and professors for taking the time to discuss concepts, provide feedback, and the constant encouragement from start to finish. None of this would be possible without all of you.

Michael T. Larkin

# Table of Contents

Page

# List of Figures

# List of Tables

A STOCHASTIC GAME THEORETICAL MODEL FOR CYBER SECURITY

# I. Introduction

## 1.1 Background

The difficulty in assessing cyber defense and resiliency is high due to the intangible nature of cyberspace and is compounded by the rapid development of technology. Cyber hardening, defined by the construction of network infrastructure to bolster security via system redundancy, detection and prevention technologies, and acceptable use policies [2], have been researched extensively with direct real-world applications and results. However cyber resiliency, or the ability of a system to deter, withstand, and recover from harmful events [2, 3], remains predominately theoretical. This does not mean that researchers are not striving to obtain actionable methods to quantify and define the resiliency of a cyber system. On the contrary, researchers have been investigating the problem for decades. However, with each method proposed, adversaries continue to develop unique strategies and methodologies that negate preemptive measures [4].

Cyber warfare has gained increased awareness over the last two decades due to hostile actions taken by nation states. What seemed liked science fiction has become a frightful reality, as seen in 2009 when adversaries launched a virus called Stuxnet, gaining access to an Iranian nuclear weapons program and enabling remote access to centrifuges that enabled the processing and enrichment of nuclear materials [5, 6]. These adversaries accomplished kinetic effects via cyberspace by adjusting settings on the centrifuges, causing them to spin beyond the functional threshold and subse-

quently destroying them [6]. This attack halted production for a year and remains unattributed to this day [5]. Stuxnet shocked cyber security analysts into reevaluating the security and resiliency of their cyber systems resulting in a shift in focus to cyber resiliency research.

The realization that actions in cyberspace have the potential to create kinetic effects reemphasizes the importance of cyber security. Game theory provides a potential solution for cyber security issues and allows researchers to evaluate the actions between attackers and defenders in cyberspace by assessing the strategies and payoffs for each side. Current methods in applying game theory for cyber defense analysis frequently assume perfect information, in that both the attacker and the defender have complete knowledge of the strategies and capabilities of the opposing player. This approach simplifies the problem and allows for a more tractable solution, yet creates a barrier in application to real-world scenarios. While each side may know their respective capabilities, intent, and strategies, there is a degree of uncertainty as to the same qualities of the opposition. Furthermore, since cyber actions progress rapidly and are difficult to detect and analyze in real-time, the degree of uncertainty increases, making an imperfect game more desirable in practice.

This research develops the use of game theoretical models in order to provide the optimal defensive response to adversarial threats in the cyber domain and to provide insights into resource allocation when improving cyber resilience. This study will cover various games with utilities that include a degree of uncertainty in the opponent's strategy. By accounting for uncertainty, additional fidelity is gained in player outcomes and provides results that better emulate reality.

## 1.2 Problem Statement

Develop a game theoretical model in which cyber defense resiliency can be quantified from the perspective of the cyber system defender. Develop action and state spaces for an attacker-defender scenario to provide a realistic evaluation of cyber resiliency.

## 1.3 Research Objectives

This research seeks to answer the following questions to quantify cyber defense resiliency and validate proposed models.

***Question 1.***

How can the resiliency of a cyber system be evaluated using game theoretical and stochastic modeling?

***Question 2.***

Given the current state of a cyber system and potential actions of a malicious actor, how may cyber security analysts evaluate the likelihood of success or failure of defensive cyber actions?

***Question 3.***

In what manner can a cost analysis be performed on an upgrade or replacement of cyber defense hardware and software while in turn evaluating the effects on the system?

## 1.4 Assumptions

First, the game theoretical model is set as a two-player non-cooperative game such that players are defined as the "Blue" force and "Red" force, indicating the defender and attacker, respectively. While each of these forces may realistically consist of multiple entities performing various simultaneous actions, the sequential actions of an attacker-defender scenario are best captured when each force is treated as a singular entity.

Second, retaliatory actions by the defender are not considered as these actions counter the objective of evaluating cyber defense resiliency. Additionally, attribution is difficult to obtain in cyberspace, making retaliation by a defensive force difficult and worthy of its own research.

Finally, data from cyber systems are not readily available due to system complexity, security risks, or a lack of understanding of what constitutes applicable data. As such, all values used in the game models are nominal. The focus of the research is on the methodology with the numerical evaluation supporting the methods used. Therefore, the subjectivity of the values do not pose a risk to the validity of the proposed models.

Additional assumptions are inherently necessary in specific types of game theoretical models and will be addressed accordingly as they are presented in the development of the research methodology.

## 1.5 Overview

This thesis consists of six chapters, to include this introductory chapter, Chapter I. Chapter II provides a literature review consisting of a brief history of cyber warfare and current research conducted utilizing game theory to evaluate cyber security.

Chapter III introduces various concepts in game theory as well as examples of how each concept can be leveraged in analyzing various scenarios common to cyber defense. Chapter IV discusses the methodology used in formulating the proposed game theoretical model. Chapter V provides an analysis of results from the model using synthetically-generated data, followed by conclusions and further research in Chapter VI.

# II. Literature Review

## 2.1 Overview

The purpose of this chapter is to define and provide the current issues surrounding resiliency as it pertains to cyberspace. This chapter consists of four sections, each discussing the history, necessary terminology, and proposed game theoretical models for interactions between an attacker and defender in cyberspace. Section 2.2 provides a brief history of cyber warfare. Section 2.3 defines and differentiates between cyber resiliency and cyber hardening. Section 2.4 provides an overview of the current challenges of cyber defense. Section 2.5 discusses the current game theoretical models proposed and utilized in providing a response to the attacker-defender scenario within the cyber domain, as well as a brief overview of game theory and its key concepts. Finally, Section 2.6 provides an overview of stochastic modeling by means of discrete time Markov chains.

## 2.2 History of Cyber Warfare

Cyber warfare has gained significant importance over the past decade as incidents increase in both frequency and severity, and new technology develops. With growing interest in computer science and related fields, research and development in both cyberspace and cyber warfare has progressed rapidly. The development and progression of cyber warfare can be categorized in four distinct phases, or insights, as defined by Warner: "Computers can spill sensitive data and must be guarded (1960s)... Computers can be attacked and data stolen (1970s)... We can build computer attacks into military arsenals (1980s and 1990s)... Others might do that to us - and perhaps already are (1990s)" [7]. These insights form the basis for the overview to follow.

The term "cyberspace" was first used in 1984 in the science fiction novel *Neuromancer* by William Gibson and became common terminology in the early 1990s, originally viewed as a component of information warfare [8, 9], or "operations carried out to defend our own information and our own information systems, or to attack and affect the information and information systems of an enemy" [10]. Cyberspace is defined as "an operational domain framed by the use of electronics and the electromagnetic spectrum to create, store, modify, exchange, and exploit information via interconnected and Internetted information systems and their associated infrastructures" [11]. It is currently viewed as its own warfare domain while remaining closely tied to information operations. Associated with cyberspace is cyberpower, defined as "the ability to use cyberspace to strategic advantage and to influence events in the other operational environments and across the instruments of power" [11]. Cyberpower is used to express the dominance of an organization or nation state in cyberspace. It is frequently used in the context of a quantitative measure, yet is subjective due to the difficulty in accurately determining a quantitative or qualitative value in which to make comparisons.

During the 1960s, computer espionage became a concern for the first time as malicious users took advantage of the open access to computer data. Users were beginning to explore the nuances of computers and the potential added utility provided in day-to-day operations. Data could be stored and shared in a digital format for the first time allowing for greater access to all users. However, it quickly became apparent that with greater access comes an increased potential for data to be stolen or manipulated for malicious purposes. These criminal actions spurred the first implementation of administrator privileges, password hashing, data encryption, and file permissions in the early 1970s [7].

The first large-scale cyber related attack occurred in 1988. Named the "Morris Worm," the attack leveraged network resources to the point of causing shutdowns of portions of the Advance Research Projects Agency Network (ARPANET), the precursor to the Internet. This development was the first major cause of public concern in cyberspace and fueled the creation of the Computer Emergency Response Team (CERT), a team constructed to focus solely on the computer security for the United States [12].

In the mid 1990s, Russia and China began recognizing the United States' focus and dominance in the cyber domain, as well as the reliance on the U.S. in obtaining their own hardware. As such, both countries began focusing on their own development of computer systems, specifically within their respective military, fearing that the U.S. may embed viruses in the computer systems that could be triggered at a later date rendering their systems useless [7].

As computer development and interest began to grown internationally, the U.S. government grew concerned with their own network infrastructure limitations. In 1995, the RAND Corporation led exercises to determine how well US systems and defensive cyber teams can respond to a cyber conflict. The results were disastrous and shocking to those acting as the defensive team. The U.S. infrastructure was shown to be less stable than previously thought and critical infrastructure could be thoroughly degraded or destroyed solely by cyber attacks [7].

Similarly, the 1997 exercise ELIGIBLE RECEIVER was executed by direction of the Joint Chiefs of Staff. Participants were divided into a "blue team" (friendly actors) and "red team"(hostile actors), where the red team "was restricted to using store-bought computers and hacking tools downloaded from the Internet" [7]. The goal of the exercise was to test how well the Department of Defense (DoD) operated with partner government branches in the midst of an active cyber event. The results

were similar to those in the RAND exercise. While the red team was not permitted to create actual effects, they could clearly prove that they had the capability to cause severe harm to the nation's critical infrastructure [7].

Coinciding with the growing local and international concerns in limitations of cyber security practices, active cyber attacks began to appear on a regular basis in the 1990s with hobbyists and script kiddies (amateur hackers who use existing scripts and codes for their own purposes), organized crime, and nation state driven reconnaissance via the Internet. Typical attacks came in the form of spyware, rootkits, bots, spam, phishing, credit card fraud, identity theft, corporate information theft, and denial of service (DoS) [11].

These techniques have not lost prevalence in the following years, but have instead gained greater severity with increased computing power and the integration of large-scale systems. By the 2000s, hostile users began to target large corporations, perform massive scale credit fraud, and implement DoS by utilizing large botnets in order to damage nation state infrastructure [11]. Additionally, NATO executed the first public use of militarized cyber warfare against Yugoslavia with the use of website defacement, the spread of propaganda via media outlets, and distributed DoS (DDoS) [12].

The major impetus for enhanced cyber security practices and awareness occurred in 2010 with the release of the Stuxnet worm. Stuxnet was released on an Iranian nuclear facility in order to damage centrifuges by attacking "Siemens' Supervisory Control and Data Acquisition (SCADA) systems that are used to control and monitor industrial processes" [12]. While this attack was never been attributed to a source, it was believed that the instigator may have been one or more nation states with a vested interest in seeing the nuclear program disrupted or destroyed. This attack fueled worldwide awareness to the criticality of cyber security research on information

systems and the potential destruction that can be left in the wake of successful actions in the cyber domain [12].

## 2.3 Cyber Resiliency and Hardening

As defined by Wilner, resilience, and cyber resilience by extension, "is the ability to bounce back, to mitigate the effects of an attack, or recover quickly after getting hit" [3]. He further states that resilience is linked with deterrence in that if the actions taken by an attacker have little to no effect on a resilient system, the attacker is less likely to begin or continue to press the attack. Thus, the attacker or would-be attacker is deterred from progressing further [3].

Leveraging a number of characteristics of resilience, Haimes provides a similar definition that encapsulates all of them: "Resilience... is defined as the ability of the system to withstand a major disruption within acceptable degradation parameters and to recover within an acceptable time and composite costs and risks" [2]. Both authors provide definitions with varying degrees of specificity but the underlying concept holds true for both: resilience is the ability of a system to resist or recover from a harmful event originating from an internal or external source.

Cyber hardening refers to building network infrastructure that bolsters security "by building fences or formulating policies and procedures that would limit access to infrastructures" [2]. This is typically done by the implementation of firewalls, intrusion detection systems, intrusion prevention systems, and a set of rules or policies to limit vulnerabilities and traffic flow to what is deemed acceptable and trustworthy. Haimes further notes that hardening provides little to no assistance in minimizing the recovery time after an event and any associated costs and risks [2].

While resiliency and hardening are related, improving one does not necessarily improve the other. From these definitions, hardening can be viewed as a sub-component

that bolsters resiliency, but not the converse. Thus, in the search for methods to improve resiliency, hardening techniques remain in consideration.

There are points of overlap between resiliency and hardening in the redundancy and robustness of the system. Redundancy is the use of additional components or systems with the goal of supplementing the primary component or system in the event that it fails [2]. From a hardening perspective, redundancy bolsters security by providing the option to isolate a failed or attacked system without adversely affecting system availability and performance for users. This same reasoning is used for improving resiliency since negative effects on the system can be minimized or negated. Haimes defines robustness as "the degree of insensitivity of a system to perturbations or to errors in the estimates of those parameters affecting the design choice" [2]. Robustness, then, hardens a system by providing a configuration in which maximizes the availability of network resources while making the system resilient by ensuring the system remains unaffected by internal and external disturbances, be it malicious or otherwise.

## 2.4 Challenges in Cyber Defense

The challenges faced in performing and evaluating cyber defense are vast. In preventing adversary actions, defenders must be able to anticipate and have complete knowledge of every flaw in their hardware and software. However, such breadth and depth of knowledge is simply infeasible as new developments are made at a rapid pace. As network architecture and associated capabilities increase in complexity, so do the associated risks [13]. The following three sections provide a brief overview of current common issues in cyber defense research: vulnerability management, attribution, and metrics.

***Network Assessment.***

A key component in determining the quality of security on any network is the frequency and accuracy of network assessments. These come in many forms and are dependent on the criteria used to measure the assessment. One fundamental form of assessment is a vulnerability rating based on the number of devices on the network that do not have the most current patches or contain known vulnerabilities. Scanners are used to make a vulnerability determination and are updated upon release of new patches and vulnerability signatures. Scans provide a good indication of how vulnerable a system may be, however they lack the ability to show how the network can resist or respond should a vulnerability be exploited.

This last assessment is difficult to determine due to the ambiguity in what defines a resilient system. An attempt is made for military networks by focusing on the stability of the network infrastructure in the event of an attack. On the surface this may seem to be an adequate approach, however it does not necessarily capture the information component of network resiliency. The military often attributes cyber attacks to some degree of destruction of the infrastructure devices without considering the information that may be lost or stolen from those same devices. As such, current practices tend to provide a "subjective and unreliable assessment of impact" [14].

***Vulnerability Management.***

Network administrators frequently experience complications in cyber security due to regular patch cycles. A patch refers to settings and updates that eliminate discovered vulnerabilities in an operating system or software. These patches are released in cycles originating from the developer on a monthly or bi-weekly basis and are frequently publicly announced. Awareness of known vulnerabilities is critical for administrators, however this awareness is also provided to potential attackers. As

patches are announced, attackers gain new paths and methods to exploit a system until administrators can take appropriate security actions.

Compounding this issue, patches are released and announced regularly on what is known as "Patch Tuesday," followed the next day by "Exploit Wednesday." The danger of regular patch cycles is noted by Jajodia: "The dynamics of this process significantly favors the attacker over the defender because the attacker needs to find only a single exploitable bug while the defender must ensure none exist" [15].

Additionally, patch management is highly dependent on the Internet for distribution to clients. While some vendors have distribution solutions in the event the Internet becomes unavailable, many vendors do not. Patches can have inherent latent issues that may not become apparent until deployed to the system. It is gradually becoming a common practice to include the criticality of each patch in releases, however it has yet to reach across all vendors. When vendors neglect to provide this information, users become hesitant to apply the patches, thereby making the network more vulnerable to attack [11].

Vulnerabilities are not strictly limited to the system and software. Human error creates additional vulnerabilities in maintaining a good cyber defense posture. Infiltration of networks is often due to a failure of a user or system administrator to adhere to proper cyber security practices. These mistakes include accidentally releasing a password to an outside source, failing to properly test a patch before applying it to the system, and the absence or unsatisfactory maintenance of network security policies [11].

Finally, issues arise from the increased dependence on the interconnectivity of systems. This is most prevalent in cyber-physical systems where high-value devices are connected via a computer network to regulate and control each phase of the system's processes. Administrators favor these systems because they allow greater control

and quicker response times, critical in the oversight of industrial control systems. However, the trade-off is greater exposure to deception and infiltration attacks since "network connections across critical infrastructures create the potential for intrusions and cascading failures that can greatly magnify the impact of a small attack" [5].

The appeal for interconnectivity also comes from a false belief that if more money is spent on infrastructure and software, cyber security posturing will improve. Pfleeger notes that "investigations by other researchers indicate that some security investments can actually decrease security simply because fixing one vulnerability sometimes enables another one" [16]. Therefore, the pace of acquisitions need to coincide with the proper implementation and evaluation of each device so that the number of additional vulnerabilities introduced to the system is minimized.

With awareness and careful execution, the majority of the aforementioned vulnerabilities can be successfully overcome. Unfortunately, even if users and administrators adhere to optimal security practices, it is impossible to account for all vulnerabilities. As an example, Wilner notes that "backdoor portals and zero-day software vulnerabilities seemingly pile up. Even air-gapping secure networks from unsecured networks - which can involve physically separating internal digital space from online digital space - is not foolproof" [3]. Regardless, reducing the attack surface by accounting for these flaws can still thwart a great deal of the most compromising and destructive attacks.

***Attribution.***

Defense becomes simplified if one knows who the adversary is and what actions they can take. In the physical space, one does not antagonize an opponent unless they can reasonably conclude that the opponent can be, at a minimum, matched in resources, technology, strategy, or sheer destructive power. Likewise, one can

predict to a degree of accuracy the chances of success in defending an attack given observations made of the opponent. Cyberspace introduces a layer of complexity where actions can be taken without being cognizant of the physical location of the source and the person or entity behind the action. Wilner highlights this fact by stating "the problem of attribution - who to blame for an attack and who to retaliate against as a result - is a knotty problem in digital space" [3]. Resources that may not necessarily be owned by the actor can be leveraged to further obscure the source. Cyberspace, then, "provides traditionally weaker states, non-state actors, collectives, and individuals disproportionate power over traditionally powerful states" [3].

Furthermore, the interconnectivity of devices via the Internet has increased the ease in which attackers can obfuscate their identity and location. Many attacks, such as DoS and DDoS, utilize the victim's network or an external network to launch an attack so that the attacker may keep their own identification information hidden behind layers of devices and Internet protocols (IPs) [11]. As a result, the victim is incapable of attributing the attack to a single source resulting in misguided root-cause analysis.

Frequently, attacks that seem as though they may not be malicious are falsely attributed to a failure with the system or the network. Pfleeger notes "an incident's cause isn't always clear - for instance, sluggish network behavior can be the result of a virus, a denial-of-service attack, or simply an unusual but benign spike in network activity" [16]. This oftentimes enables advanced-persistent threats, attacks that occur over an extended length of time, to thrive undetected for months or years [3].

***Metrics.***

A source of contention in metric standardization is that metrics are dependent on the research being performed and what cyber phenomena is to be captured from

said research. For example, when evaluating intrusion detection one may consider bandwidth a critical metric since attacks across a network may cause packet drops and a degradation of user connectivity. However, attacks exist in which bandwidth remains unaffected or contains negligible changes such that it appears as typical daily network traffic. In this case, bandwidth may not be a critical metric or may only be considered alongside sensor data from an intrusion detection system (IDS) or firewall. One may continue to add elements into the evaluation, however without standardization it may not be clear when sufficient data has be obtained and the value of each metric in the overall evaluation.

Capturing data introduces a new layer of difficulty. The integrity and validity of the data must first be considered. Once the data is deemed valid and free of data corruption, then the required amount of data and its source, that is from a test environment or live system, must next be considered. Furthermore, should data be taken from a test environment, a determination must be made on how closely the environment resembles the real-world system. Because of the unique requirements imposed on a live system, it may not be possible to construct a similar test environment to the degree that is required.

These are only a few of the decisions to be made for appropriate metrics, however "no proposed set of metrics is universally accepted or embraced as useful, and no framework lets organizations answer their wide variety of questions about network and information security" [16]. To correct the lack of agreement, Pfleeger argues that "we must address two problems: selecting attributes that reflect the cybersecurity aspects of interest and finding appropriate ways to combine these attributes so that we can measure overall cybersecurity" [16]. These two problems are certainly nontrivial due to the adaptive nature of attackers and the cyber domain, statistical analysis

is very difficult. Additionally, not every parameter can be captured since there are far too many [17].

Kramer *et al.* [11] propose that a basis for measurement in performance is "connectivity, availability, and bandwidth". This provides a solid foundation, however "measurements are often made infrequently, inconsistently, and incompletely, frustrating those who want to use the results" [16]. Without the availability of data, the measurement debate may remain unresolved.

Once standardized metrics are agreed upon, the next logical step is to relate models constructed within a test environment to real-world systems. Gaining metrics from attack testing is difficult since attackers frequently modify their attacks to increase severity and effectiveness [18]. The number of potential modifications are vast and dependent on the systems utilized by both the attacker and the defender, making enumeration and testing a seemingly intangible objective. Additionally, hackers and penetration testers develop new techniques regularly to infiltrate and exploit systems, making it near impossible to project the possible actions taken by an attacker. This uncertainty "is the unique and perhaps the biggest uncertainty in real-time security analysis" [17]. However, malicious actors are also subject to a degree of uncertainty as even the most skilled attacker encounters unforeseen problems causing a failure in their action. As Xie *et al.* [17] note, "Cyber attacks are not always guaranteed to succeed, thus there is the uncertainty from the imperfect nature of exploits".

## 2.5 Game Theory in Cyber Defense

Game theory provides a mathematical model of the interactions between players and has gained increased prevalence due to its logical applicability to a vast range of topics, to include economics, computer science, biology, and politically, as well as gaining additional traction in cyber security research [19, 20]. In its fundamental

form, game theoretical models define a set number of participants, called players, who are each defined by a set of actions and strategies. These actions and subsequent strategies make the critical assumption that players act rationally, that is "a perfectly rational player could justifiably play [a strategy] against one or more perfectly rational opponents" [21]. Games may be constructed as cooperative, or coalitional games, where players work together to maximize the utility for all players, or non-cooperative games where each player seeks to maximize their own utility.

Additionally, each player is assigned a utility, or payoff, based on the action or strategy played. These utilities can take many forms depending on the context of the game, be it monetary, probabilistic, or synthetically generated values, for example. The action space and utilities for each player can be affected depending on the knowledge each player has of the opponent's action. Perfect information occurs when each player has knowledge of every decision made by all other players, as opposed to imperfect information where some or all of the decisions made by the other players is unknown. This is not to be confused with complete and incomplete information. Complete information occurs when each player knows the number of players in the game, their respective strategies, and the utility functions for each player, whereas incomplete information occurs when there is uncertainty about any or all of this information [1, 4].

Games can be expressed in normal form or extensive form. Normal form games restrict each player to act simultaneously and are expressed in the form of a matrix. Extensive form games add a temporal aspect to the game over potentially several turns vice requiring simultaneous player actions over a single turn as found in normal form games. Actions are taken by players sequentially over either an infinite amount of time or until a terminal point is reached. Each player's action space is determined by what is feasible and rational for that player based on the action of the previous

player and the completeness of knowledge of that action. Extensive form games can be represented in tree form, as found in sequential games, but can also be shown in normal form where each column or row indicates the strategy, or set of actions, taken by the respective player.

To fully understand these concepts, it is typical to examine the "Prison's Dilemma" game. This game is a normal form non-cooperative game consisting of two players, A and B, both of whom are suspects of a crime. Both suspects are being detained by the police and questioned separately in order to determine who is guilty of the crime. Each of the suspects can take one of two actions, defect by accusing the other of the crime or cooperate by saying nothing. The actions taken by each player determines the number of years each will spend in prison. If A and B both defect, they will both serve a sentence of three years, whereas if they both cooperate they will both serve a sentence of one year. Additionally, if A defects and B cooperates, A will not serve any time in prison while B will be sentenced to five years. The same applies for the converse situation with the payoffs applied similarly. Since each suspect desires the shortest sentence possible, each utility is expressed as a negative value. For example, a sentence of one year is expressed as -1 [1].

The normal form matrix is shown in Table 1. The utilities are read as pairs where the first value corresponds to A and the second with B.

| | | B | |
|---|---|---|---|
| | | Defect | Cooperate |
| A | Defect | $-3, -3$ | $0, -5$ |
| | Cooperate | $-5, 0$ | $-1, -1$ |

**Table 1. Prisoner's Dilemma Normal Form Matrix [1]**

To find the solution to this game, the best payoff for each player must be found such that neither player can do any better by deviating from their strategy. This means that given one player chooses to play a specific strategy, the other player chooses the best response to that strategy such that their utility is at least as good as all other responses. Thus, the best response need not be unique. The strategy played is known as the Nash equilibrium. The Nash equilibrium may not necessarily be unique either and may not exist. However, in a mixed strategy game where each player's strategies are played probabilistically, at least one Nash equilibrium will always exist. Therefore, if a Nash equilibrium does not exist in the normal form game, one may turn to a mixed strategy game in order to find an equilibrium.

Returning to the Prisoner's Dilemma and first considering A's best response to B, it is shown that B receives the best payoff when choosing to defect with a sentence of zero years. Given this strategy by B, A's best response is to defect as well since the sentence of three years is shorter than the sentence of five years if A cooperates. Using the same logic to find B's best response to A, it is found that A will defect. Therefore, a unique Nash equilibrium is found with both suspects defecting and each receiving a sentence of three years. While this may be a surprising result since each suspect can receive a better payoff by cooperating, neither suspect can reasonably expect the other to choose to cooperate since the best payoff comes from defecting, regardless of whether or not the suspects are allowed to discuss before being questioned.

Extensive research currently exists in the application of game theory for cyber security, however it is predominantly theoretical with few exemplars given. The primary game theoretical models used in research today for cyber security are those of the extensive form, stochastic, and Bayesian games.

Yuan *et al.* [22] present a two-level Stackelberg game in order to evaluate the actions taken by a resilient control system as part of a cyber-physical system in the

event of a denial of service or distributed denial of service attack. Each level represents the layer of the cyber-physical system in which each Stackelberg game is being played. The first, or internal, level takes place at the cyber layer where the players are the intrusion detection system (IDS) and the cyber attacker. The second, or external, level occurs at the physical layer between the external disturbance on the system and the controller [22].

The Stackelberg game is played where one player acts as the leader and the other the follower. The leader is a dominant position as it can enforce strategies on the follower, giving the leader a first-mover advantage. In this instance, the leader and follower are defined as they are previously listed in their respective level. The cost functions for each level are represented with respect to the leader, where the first level seeks to minimize the cost of a false alarm for the IDS and the second seeks to maximize the amount of disturbance caused by the attacker.

In their analysis, Yuan *et al.* [22] found that a unique Stackelberg equilibrium can be found from their theoretical two-level model. Their work has been applied to a power grid system furthering the efficacy of their proposed model [22].

Stochastic games build on a normal form game by including a state space in which the game will move between states given a specific transition probability and the actions taken by each player. The payoff for each player is based on what state the game is in at the termination point [21].

In their paper on attacker strategies and finding the optimal defensive response strategy, Jiang *et al.* [23] propose a two-player, zero-sum stochastic game where the players are defined as the attacker and defender. The state space is defined by the level of system privileges the attacker has obtained at any given point in the game with transitions occurring based on the action taken by the attacker. Since the attacker is continuously gaining privileges from the defender's system, the utilities for

each player are additive inverses based on the amount of cost to the defender that considers operational costs, cost of damage, any residual costs, and the cost effect on the system based on the defender's action [23].

The authors conclude that their theoretical approach can successfully predict the attacker's strategy and apply an optimal defensive response strategy. Additionally, they acknowledge that there are limitations in their model. They first note the difficulty in evaluating all the attack strategies that will allow the attacker to escalate privileges. Second, the authors note that in order to test their model, they were required to manually provide values for all parameters considered [23]. This echoes a common issue in analysis of cyber defense where good usable data is severely lacking.

A variation on the stochastic game, the Bayesian game, is used by Ouyang *et al.* [24] in order to evaluate a scenario in which there is asymmetric information amongst players and strategies are updated as the state of the system changes. Games with asymmetric information are those in which "agents have different information over time" [24]. The authors introduce the concept of a common information based perfect Bayesian equilibria which "consists of a pair of strategy profile and a belief system that are sequentially rational and consistent" [24]. They further highlight the fact that typically a Nash equilibrium in these types of games would be one which does not account for deviations in the stage-to-stage progression, only deviations made by the players at the outset of the game. Thus, they require that any equilibria found for this game must be those such that players can make no better deviation at any stage of the game, not just the start.

Ouyang *et al.* [24] show that their model provides an equilibrium for a specified subset of dynamic games. In their research they define the existence of common information based perfect Bayesian equilibria and provide a solution method by

backward-induction. Finally, they utilize their method in a small example to show the calculations of each stage of the game and its progression.

Kovach *et al.* [25] discuss the use of hypergame theory in various applications and note that there is currently limited research using this technique for cyber defense strategies. Hypergame theory is an extension of game theory that includes the value of the beliefs and perceptions of each player into the formulation of the model. While no specific model is proposed, it is noted that this approach allows for the inclusion of deception based attacks, which are often a concern when evaluating the trustworthiness and reliability of outputs from a cyber-physical system. Additionally, hypergame theory enables researchers to take into consideration the rationality and intent behind an adversary's actions, for which neither game theory nor decision theory take into account [25]. Utilizing hypergame theory may help improve evaluations of attacker-defender scenarios as researchers discover more information as to what drives attackers to act in a hostile manner and how a defender should respond. The matter of intent is quickly increasing in relevance since, as Connell notes, "adversaries - whether state or non-state actors - are likely to view interactions in cyberspace very differently than we do" [26].

## 2.6 Discrete-Time Markov Chains

This section provides an overview of key concepts for discrete-time Markov chains using concepts and notation as presented in [27] and [28].

A stochastic process is defined by $\mathbf{X} = \{X(t), t \in T\}$, where $X(t)$ is a random variable that represents the state of the stochastic process over a defined time $t$. These states $X(t)$ can represent a number of things from weather status and countries in conflict at time $t$, to the number of customers processed by tellers at a bank at time $t$. If $T$ is countable, the stochastic process is called a discrete-time stochastic process.

Otherwise, the process is a continuous-time stochastic process. The state space is then defined as the collection of possible states represented by $X(t)$ [27, 28]. Since this research considers only discrete-time stochastic processes, continuous-time stochastic processes will not be discussed further.

For a stochastic process $\mathbf{X} = \{X_n, n = 0, 1, 2, \ldots\}$ and finite $n$, when $X_n = i$ the stochastic process is defined to be in state $i$ at time $n$. Let $P_{ij}$ be the probability of moving, or transitioning, from state $i$ to state $j$ and

$$P\{X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \ldots, X_1 = i_1, X_0 = i_0\} = P_{ij} \tag{1}$$

for states $i_0$, $i_1$, ..., $i_{n-1}$, $i$, and $j$ and $n \geq 0$ [27]. Thus, "any future state $X_{n+1}$, given the past states $X_0$, $X_1$, ..., $X_{n-1}$ and the present state $X_n$, is independent of the past states and depends only on the present state" [27]. This stochastic process defines a Markov chain. If $n$ is finite, the chain is said to be a discrete-time Markov chain. Furthermore, the probabilities of transitioning from state $i$ to state $j$ can be represented as a transition matrix $\mathbf{P}$ for all $i, j \geq 0$ [27, 28].

$$\mathbf{P} = \begin{pmatrix} P_{00} & P_{01} & P_{02} & \cdots \\ P_{10} & P_{11} & P_{12} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ P_{i0} & P_{i1} & P_{i2} & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \tag{2}$$

Since $\mathbf{P}$ is consists of probabilities, the following must hold true:

$$\begin{aligned} P_{ij} &\geq 0, \quad i, j \geq 0 \\ \sum_{j=0}^{\infty} P_{ij} &= 1, \quad i = 0, 1, 2, \ldots \end{aligned} \tag{3}$$

Additionally, if $P_{ii} = 1$ for state $i$, the state $i$ is an absorbing state since once the state is entered, it is never left again. Thus by Equation 3, for absorbing state $i$ and state $j \neq i$, $P_{ij} = 0$.

A Markov chain may be represented by a set of classes consisting of states that communicate with one another. If two states $i$ and $j$ communicate, then $P_{ij} > 0$ and $P_{ji} > 0$. If only one class exists for a Markov chain, that is all states in the chain communicate, it is said to be irreducible [28]. Chains can be further described as periodic with period $d$ "if $P_{ii}^n = 0$ whenever $n$ is not divisible by $d$ and $d$ is the greatest integer with this property" [27]. Otherwise, if a chain is both irreducible and not periodic, that is a state has a period of 1, it is said to be aperiodic [27, 28].

States are further classified into being either transient or recurrent. As defined by Ross [28], "For any state $i$ we let $f_i$ denote the probability that, starting in state $i$, the process will ever reenter state $i$. State $i$ is said to be *recurrent* if $f_i = 1$ and *transient* if $f_i < 1$." In other words, a state is considered recurrent if it is guaranteed that the chain will return to the state, and transient if there is a probability that a state will be left without being reentered again. Note that all absorbing states are recurrent, but the converse is false.

Recurrent states are further defined by the expected number of transitions made before returning to a state by being positive recurrent or null recurrent. Letting $m_j$ be the expected number of transitions before returning to state $j$, Ross [28] defines each as "the recurrent state $j$ is *positive recurrent* if $m_j < \infty$ and say that it is *null*

*recurrent* if $m_j = \infty$." Using these and previous definitions, an ergodic Markov chain is defined as one that is positive recurrent, irreducible, and aperiodic [27].

These state classifications give way to finding the long-run proportions for states in a Markov chain, or the proportion of transitions occurring between states. Letting $\pi_j$ be the long-run proportion for state $j$ and $\pi_i$ be the long-run proportion of state $i$, each $\pi_j$ is found by solving the following system of linear equations [28]:

$$\begin{aligned} \pi_j &= \sum_i \pi_i P_{ij} \quad j \geq 1 \\ \sum_j \pi_j &= 1 \end{aligned} \tag{4}$$

The first equation shows the long-run proportion of state $j$ given an initial, or starting, state $i$ and the respective transition probability. Since each $\pi_j$ is a proportion, the second equation must hold true for all states $j$. Furthermore, if a Markov chain is ergodic, then a unique limiting distribution is found by

$$\begin{aligned} \pi_j &= \lim_{n\to\infty} P_{ij}^n = 0, \quad \text{if state } j \text{ is transient or null recurrent} \\ \pi_j &= \lim_{n\to\infty} P_{ij}^n > 0, \quad \text{if state } j \text{ positive recurrent} \end{aligned} \tag{5}$$

and equals the long-run proportion for state $j$ [27, 28]. This distribution is also said to be stationary, defined by [27]

$$P_j = \sum_{i=0}^{\infty} P_i P_{ij}, \quad j \geq 0 \tag{6}$$

Letting $T = \{1, 2, \ldots, t\}$ be the set of transient states, a matrix can be formed showing the transition probabilities between each of the transient states. Since, the matrix contains only transient states, the sum of each row may no longer be equal to one. Thus, the transient state matrix is represented by [27]

$$\mathbf{Q} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & \cdots & P_{1t} \\ p_{21} & P_{22} & P_{23} & \cdots & P_{2t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{i1} & P_{i2} & P_{i3} & \cdots & P_{it} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{t1} & P_{t2} & P_{t3} & \cdots & P_{tt} \end{pmatrix} \tag{7}$$

Leveraging the transient state matrix and $m_{ij}$ as defined earlier, the matrix of the expected number of time periods in state $j$ given it is entered from state $i$ is

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & \cdots & m_{1t} \\ m_{21} & m_{22} & m_{23} & \cdots & m_{2t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{i1} & m_{i2} & m_{i3} & \cdots & m_{it} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{t1} & m_{t2} & m_{t3} & \cdots & m_{tt} \end{pmatrix} \tag{8}$$

and represented with the equation $\mathbf{M} = \mathbf{I} + \mathbf{Q}\mathbf{M}$, where $\mathbf{I}$ is an identity matrix of size $t \in T$. Using linear algebra, this equation is equivalent to $\mathbf{M} = \left(\mathbf{I} - \mathbf{Q}\right)^{-1}$, providing values for each $m_{ij} \in \mathbf{M}$. From $\mathbf{M}$ and letting $f_{ij}$ be the probability that state $i$ ever transitions to state $j$, $m_{ij}$ is represented in terms of $f_{ij}$ by conditioning on the probability of the chain ever entering $j$ given it starts in $i$, shown by [27]

$$\begin{aligned} m_{ij} &= E[\text{number of transitions into state } j | \text{start in } i] \\ &= m_{jj} f_{ij} \end{aligned} \tag{9}$$

Therefore, the probability of ever entering state $j$ given that the chain starts in state $i$ is $f_{ij} = \frac{m_{ij}}{m_{jj}}$ [27].

## 2.7 Summary

Cyber defense and resiliency is far from a trivial problem. In its relatively short history, there have been vast technological improvements in cyberspace allowing for greater usability and integration with the added trade-off of risk due to malicious actors and behavior. Researchers continue to develop techniques to thwart attackers and minimize risk to critical systems, but the ever changing cyber environment and lack of available data continue to make research difficult to implement in real-world systems. However, the use of game theory provides the potential for gaining great insight into the future solutions to cyber defense and continues to gain traction.

While there have been a number of successes in proposed game theoretical models for cyber defense, these models are largely theoretical. A repeating theme in cyber defense research is the struggle to capture adequate data, thus many researchers must resort to manually generated data to show the validity of their proposed model. Additionally, current research tends to focus on a specific attack type or sequence vice generalizing for any attacker-defender scenario and defender network architecture. This provides sufficient results for the specified attack method but loses scalability and, potentially, relevance as attackers develop new strategies and methodology.

An additional shortcoming in much of the cyber defense research is a common underlying assumption of perfect information. This is a safe approach and allows for more feasibility in finding an appropriate model, however does not treat itself well to real world scenarios. As recent attacks have shown, it is not a trivial matter for a defender to detect and identify an attacker. Because of the signature based operation of defensive hardware like IDSs and firewalls, an attacker can bypass these devices if

they can avoid using an attack strategy that resembles any of the known signatures. If an attacker is successful, the odds of detecting the infiltration and exploitation become increasingly worse for the defender. Therefore, cyber defense lends itself more appropriately to the use of imperfect and, possibly, incomplete information. However, while more appropriate, this only increases the difficulty in modeling an attacker-defender scenario. If realistic data were to become available showing effects from actions taken by both the attacker and defender, game theoretic modeling can increase the fidelity of capturing the real world interactions.

Current research tends to focus efforts on models in which a single technique is used. In turn, these methods often lack breadth and depth of analysis of the overall cyber security assessment. This research departs from this commonality by encompassing multiple techniques to gain greater fidelity in the current resilience of a cyber network and the recommended steps for improvement.

Additionally, this research uses notional data as previous researchers have done due to the lack of real-world data availability. While imperfect information is not directly addressed, uncertainty is introduced via the use of stochastic modeling. The start of the proposed model makes assumptions on the starting state of the system, introducing elements of imperfect information, but has the potential to become more refined by eliminating degrees of uncertainty as information is gained during the game progression. Furthermore, the model resembles imperfect information by considering all possible actions and system states for the defender and attacker. Finally, should real-world data become available, the proposed model can quickly provide real-time assessments on the game based on the current state of the system.

# III. Applications in Cyber Security

## 3.1 Overview

This chapter provides examples of foundational game types within the context of cyber related scenarios as well as their solution methods. The game types and solution methods herein form a basis for solving the model outlined in this research. First, normal form games are addressed in both the general sum and zero-sum forms with pure-strategy Nash equilibria. Next, the normal form games are revisited to show how mixed-strategy Nash equilibria are found so each player can play randomized strategies. This is followed by a minmax game where one player wishes to find a strategy that results in the smallest maximum expected utility for the opponent. Finally, the extensive form game provides a scenario in which the interactions between players may be viewed in an explicit temporal structure as opposed the simultaneous actions of players found in normal form games [21]. All scenarios and data points are notional and not based on actual events.

## 3.2 Normal Form

Software and hardware issues are frequently identified post commercial release. In response, developers create fixes, called patches, that are released to users for deployment. Software development companies like Microsoft with a large user base have adopted regular releases of new patches, known as "Patch Tuesday" [29].

On the second Tuesday of each month, patches are publicly advertised via Microsoft Security Bulletins. It is the responsibility of system administrators to be aware of the current vulnerabilities addressed by the patches, how to appropriately deploy them, and the potential affect to their system. The day following Patch Tuesday is "Exploit Wednesday," a tongue-in-cheek name as patches are released publicly

allowing attackers to become aware of existing vulnerabilities and have a window of opportunity to attack vulnerable networks.

Microsoft provides a rating scale for each patch based on a number of factors, including exploitability, components affected, and degree of expected compromise. The ratings from lowest to highest are Low, Moderate, Important, Critical (an explanation of the rating system can be found in [30]). A knowledge base (KB) number system is used to identify each patch in the repository. For example, two patches announced on the March 2017 security bulletin are KB4013242 - Security Update for Microsoft Exchange Server rated as Important, and KB4013073 - Cumulative Security Update for Internet Explorer rated as Critical [31].

To demonstrate the applicability of normal form games in patch management, suppose two system administrators, each independently responsible for Exchange for email services and web services for online services such as Internet Explorer, are planning to deploy the two aforementioned patches. Local patch management policy dictates that patches will be deployed in one of two weeks available in the patch window. Each week encompasses the required testing, progressive deployment, and problem resolution prior to the start of the following week. An integer value is assigned for each vulnerability rating in which the largest value provides the greatest payoff: Low = 1, Moderate = 2, Important = 3, Critical = 4. The leadership team has decided that it is not preferred that both Exchange and web services be patched in the same week. Due to the large user base for both services and observations from previous deployment cycles, it is desired to avoid the potential for problems with each service occurring simultaneously.

From these policies, payoffs are assigned. First, deploying both patches in the same week incurs a penalty of one-half the base payoff value. Similarly, postponing the deployment of a patch leaves the network vulnerable to exploitation for an extended

period of time, resulting in a penalty to the base payoff value. Nonetheless, postponing a patch by one week is a more favorable approach and incurs a lesser penalty of one-third of the base payoff value. In the event that both patch deployments are postponed and concurrent, the penalty of postponement is incurred first, then further reduced by the penalty due to concurrent deployment. For example, if both patches were to be deployed in the second week, the payoff of to both players would first be reduced by one-half, then reduced further by one-third of the new payoff.

Let player 1 denote the Exchange system administrator, player 2 denote the web services system administrator, and $N$ be the set of players where $N = \{1, 2\}$. Additionally, let $W$ be the numbered week of patch deployment such that $W = \{1, 2\}$. The action set, $A$, is defined as $A = \{a_j^i : i \in N, j \in W\}$. Note that $-i$ is used to denote the remaining players in $N$ such that $-i \neq i$. The utilities functions dependent on the week each player deploys patches are the following:

$$u_i(a_1^i, a_2^{-i}) = u_i(a^i) \tag{10a}$$

$$u_i(a_1^i, a_1^{-i}) = \frac{1}{2}u_i(a^i) \tag{10b}$$

$$u_i(a_2^i, a_1^{-i}) = \frac{2}{3}u_i(a^i) \tag{10c}$$

$$u_i(a_2^i, a_2^{-i}) = \frac{1}{3}u_i(a^i) \tag{10d}$$

From these utilities, the normal form game matrix is formed.

**Table 2. Normal Form Game Matrix**

| | | Player 2 | |
|---|---|---|---|
| | | 1 | 2 |
| Player 1 | 1 | 1.5, 2 | 3, 2.66 |
| | 2 | 2, 4 | 1, 1.33 |

Since $u_2(a_1^2, a_2^1)$ provides the greatest payoff for the web services system administrator, the Exchange system administrator's best response is $BR_1(a_1^2) = a_2^1$. Likewise, $u_1(a_1^1, a_2^2)$ provides the greatest payoff for the Exchange system administrator, so the web services system administrator's best response is $BR_2(a_1^1) = a_2^2$. Therefore, the two pure-strategy Nash equilibria are $(1, 2)$ and $(2, 1)$.

### 3.3 Zero Sum

Wireless technology has become a commonality in businesses allowing for greater accessibility for employees and customers alike. However, wireless devices have security limitations if access points are not carefully configured and maintained. One method used in locating and exploiting access points is war driving, a scanning technique used to capture and evaluate packets being transmitted between wireless access points and devices for the purpose of requesting network access. Attackers utilize this scanning technique in order to obtain the Extended Set Identifier (ESSID), Internet Protocol (IP) address, and Media Access Control (MAC) address of access points. Any combination of this information provides a potential target and vector for network exploitation. The following scenario demonstrates how the interaction between attackers and defenders can be represented as a zero-sum game.

Suppose a cyber security analyst is hired by a company interested in improving network architecture and configurations. The analyst finds that the devices in use have not been properly maintained and identifies issues in the wireless router. Fellow analysts have stated that a number of amateur hackers are known to practice war-driving techniques in the vicinity of the company.

War driving consists of an attacker using active scanning, traffic sniffing, or forced deauthentication. Active scanning is performed by sending probe packets to nearby wireless access points and gaining information from the response. Using the tool Net-

Stumbler, an attacker can send out a Dynamic Host Configuration Protocol (DHCP) request to obtain an IP from the victim network. The disadvantage of this method is that it is very noisy, making it easier for monitoring devices and access points to recognize the flood of probe requests and create an alert for administrator action [32].

Traffic sniffing is a passive scanning method that gathers information from access point beacon packets instead of probing. The same information can be gathered as with war driving without the disadvantage of triggering an alert from monitoring devices [32].

Finally, forced deauthentication uses traffic sniffing to obtain a MAC address from the access point's beacon packets. Once found, the attacker can send a deauthenticate message using the spoofed MAC address of the access point (that is, using the found MAC address as their own) to the broadcast address of the network, causing connected devices to reauthenticate with the spoofed MAC address. When devices try to reauthenticate, they will attempt to connect to the attacker's device, in turn providing the ESSID information to the attacker [32].

In response to these scanning methods, a defender can implement configurations on access points. First, the access point can be configured to ignore probe requests, negating the value gained from active scanning. Next, access points can be configured to omit the ESSID from beacon packets, ensuring that any device that is listening for beacons will not be able to gain the ESSID. The attacker may still be able to obtain the MAC or IP address of the access point, but information gained is less valuable without the ESSID. Finally, WPA2 authentication can be implemented requiring users to authenticate with a password versus the default setting of authenticating with the MAC address. In reality, WPA2 is the standard authentication method, however for illustration purposes it is assumed this is not the case [32].

The actions for the attacker are defined as Active Scanning (1), Traffic Sniffing (2), Forced Deauthentication (3). The defender's actions are defined as Ignore Probe Requests (1), Omit ESSID From Beacon Packets (2), and WPA2 Authentication (3). Let player 1 be the Defender, player 2 be the Attacker, and $N = \{1, 2\}$ be the set of players. Let $\delta$ be the set of defender actions where $\delta = \{1, 2, 3\}$ and $\alpha$ be the set of attacker actions where $\alpha = \{1, 2, 3\}$. Additionally, let $A = \{A_1, A_2\}$ such that $A_1 = \{a_i^1 : i \in \delta\}$ and $A_2 = \{a_j^2 : j \in \alpha\}$. Utilities are defined as the probability that the defender can defend against the attacker's action, such that $u_1(a_i^1, a_j^2) = -u_2(a_i^1, a_j^2)$. Note that all probabilities provided are nominal.

Setting an access point to ignore probe requests negates active scanning, however it serves no purpose in countering the remaining two scanning methods. Omitting the ESSID from beacon packets is useful in masking the ESSID from the attacker but does not eliminate the remaining information gathered, which may prove useful should the attacker discover the ESSID via other means, namely forced deauthentication. However, since active scanning creates more noise than traffic sniffing, there is greater risk with lower payoff. Finally, implementing WPA2 for authentication is powerful in keeping the attacker out of the wireless network, specifically in the event that forced deauthentication is attempted. This does not prevent the attacker from gathering information about the defender's network via active scanning and traffic sniffing. Regardless, these two methods are less useful if the attacker cannot gain access to network [32]. The resulting game matrix is shown in Table 3.

First, observe that $u_2(a_i^1, a_1^2) < u_2(a_i^1, a_2^2)$. Thus, $a_1^2$ is strictly dominated and can be removed since the attacker will never play this strategy. Similarly, $u_1(a_2^1, a_j^2) < u_1(a_3^1, a_j^2)$ and $a_2^1$ is removed. The reduced matrix is shown in Table 4.

**Table 3. Zero-Sum Game Matrix**

| | | Player 2 | | |
|---|---|---|---|---|
| | | $a_1^2$ | $a_2^2$ | $a_3^2$ |
| Player 1 | $a_1^1$ | $1, -1$ | $-1, 1$ | $-1, 1$ |
| | $a_2^1$ | $0.7, -0.7$ | $0.6, -0.6$ | $-1, 1$ |
| | $a_3^1$ | $0.9, -0.9$ | $0.8, -0.8$ | $1, -1$ |

**Table 4. Reduced Zero-Sum Game Matrix**

| | | Player 2 | |
|---|---|---|---|
| | | $a_2^2$ | $a_3^2$ |
| Player 1 | $a_1^1$ | $-1, 1$ | $-1, 1$ |
| | $a_3^1$ | $0.8, -0.8$ | $1, -1$ |

In the reduced form, it is clear that $u_1(a_1^1, a_j^2) < u_1(a_3^1, a_j^2)$ and $u_2(a_i^1, a_3^2) < u_2(a_i^1, a_2^2)$. Therefore, $a_1^1$ and $a_3^2$ are strictly dominated and are removed. Thus, the strict Nash equilibrium is $(a_1^1, a_2^1)$, or (WPA2 Authentication, Active Scanning).

### 3.4 Mixed Strategy

Suppose security analysts reassessed the probability of a successful defense from the previous scenario and found that the actual probability of success by omitting the ESSID from beacon packets is 0.8 against traffic sniffing. The revised game matrix is shown in Table 5.

**Table 5. Revised Zero-Sum Game Matrix**

| | | Player 2 | | |
|---|---|---|---|---|
| | | $a_1^2$ | $a_2^2$ | $a_3^2$ |
| Player 1 | $a_1^1$ | $1, -1$ | $-1, 1$ | $-1, 1$ |
| | $a_2^1$ | $0.7, -0.7$ | $0.8, -0.8$ | $-1, 1$ |
| | $a_3^1$ | $0.9, -0.9$ | $0.8, -0.8$ | $1, -1$ |

With weak domination occurring between the defender actions $a_2^1$ and $a_3^1$, there is a potential for a mixed strategy played by at least one of the players as opposed to the pure strategy Nash equilibria found in the previous game iteration. The mixed strategy of the attacker is found first.

Let $P(a_1^2) = p_1$, $P(a_2^2) = p_2$, and $P(a_3^2) = 1 - p_1 - p_2$. Assigning these probabilities makes the defender indifferent in his strategies. Solving for $p_1$ and $p_2$ using the expected payoff for the defender is performed as follows:

$$\begin{aligned} Eu_1(a_1^1) &= Eu_1(a_2^1) \\ p_1 - p_2 - (1 - p_1 - p_2) &= 0.7p_1 + 0.8p_2 - (1 - p_1 - p_2) \\ p_1 &= -1.06p_2 \end{aligned} \tag{11}$$

Either $p_1 < 0$ or $p_2 < 0$, contradicting the fact that both values must be between zero and one. Therefore, the probability of one of the strategies must be zero. Letting $p_1 = 0$, $p_2 = 0.5$, and $(1 - p_1 - p_2) = 0.5$, the expected utilities for the attacker for each of the defender's actions are $Eu_2(a_1^1) = 1$, $Eu_2(a_2^1) = 0.1$, and $Eu_2(a_3^1) = -0.9$. Note that these utilities weakly dominate $a_1^2$. Thus, it is not a reasonable strategy for the attacker and can be removed by dominance of mixed strategy. The reduced matrix can now be solved.

$$\begin{aligned} Eu_1(a_2^2) &= Eu_1(a_3^2) \\ 0.8p_2 - (1 - p_2) &= 0.8p_2 + (1 - p_2) \\ p_2 &= 1 \end{aligned} \tag{12}$$

Therefore, the attacker's mixed strategy is $(0, 1, 0)$.

Since it is guaranteed that the attacker will play $a_2^2$, the defender's actions can be evaluated easily by observing that $a_1^1$ is strictly dominated by both $a_2^1$ and $a_3^1$ given

the attacker's mixed strategy and can be removed. Additionally, the utilities for the two remaining defender actions are equal, indicating the mixed strategy for the defender is $(0, 0.5, 0.5)$. Therefore, the two pure-strategy Nash equilibria are $(a_2^1, a_2^2)$ and $(a_3^1, a_2^3)$, with a mixed strategy Nash equilibrium of $\{(0, 0.5, 0.5), (0, 1, 0)\}$.

### 3.5 Minmax Game

Information technology acquisitions is a difficult process requiring a balance of cost effectiveness, system and program requirements, and sustainability. Each gain in one may come at the cost of another, so there is risk involved in each acquisition. The following is a scenario in which a minmax game can inform the acquisition process.

An acquisitions team assesses two devices to purchase: device one and device two. Reports indicate that an increase in two types of attacks have been observed on their network: attack one and attack two. In addition to the purchase cost of each device, the team must consider the cost of repair or component replacement dependent on the resiliency of the device against each attack.

The first device has a purchase cost of \$7,000. If the device encounters attack one, an additional cost of \$1,000 is incurred and a cost of \$4,000 for attack two. The estimated cost of device one per attack is \$8,000 and \$11,000, respectively. The second device has a purchase cost of \$5,000. The additional cost from attack one is \$4,000 and \$2,000 from attack two. The estimated cost of device two per attack is \$9,000 and \$7,000, respectively.

Let player 1 denote the defender (or acquisitions team), player 2 denote the attacker, and $N$ be the set of players such that $N = \{1, 2\}$. Let $\delta$ be the set of defender's devices considered for purchase where $\delta = \{1, 2\}$ and $\alpha$ be the set of attacker's attack types where $\alpha = \{1, 2\}$. The action set, $A$, is defined by $A = \{A_1, A_2\}$ where $A_1 = \{a_i^1 : i \in \delta\}$ and $A_2 = \{a_j^2 : j \in \alpha\}$. The set of strategies, or the probability

of playing each action, is defined by $S = S_1 \times S_2 = \{(s_1(a_i^1), s_2(a_j^2)) : a_i^1 \in A_1, a_j^2 \in A_2, s_1(a_i^1) \in S_1, s_2(a_j^2) \in S_2\}$ where the sum of all strategies utilized by a player sum to one. Let $v_i$ be the base cost for device $i$ and $d_{i,j}$ be the additional cost incurred to device $i$ from attack $j$. Thus, the defender's utility function is $u_1(a_i^1, a_j^2) = v_i + d_{i,j}$. The Attacker's utility is only based on the damage done to each device. Thus, the attacker's utility function is $u_2(a_i^1, a_j^2) = d_{i,j}$. The game matrix is shown in Table 6 with utilities expressed in thousands of dollars.

**Table 6. Acquisitions Normal Form Game Matrix**

| | | Player 2 | |
|---|---|---|---|
| | | $a_1^2$ | $a_2^2$ |
| Player 1 | $a_1^1$ | $8, 1$ | $11, 4$ |
| | $a_2^1$ | $9, 4$ | $7, 2$ |

The defender seeks to minimize the maximum cost of damages inflicted by the attacker by playing a strategy in which the attacker payoff tends to zero. Alternatively, the attacker seeks to maximize the minimum cost of device purchase and damages to the defender. The minmax strategy and value for the defender are found in the following manner:

$$\begin{aligned}
&\max_{S_2}[s_1(a_1^1)s_2(a_1^2) + 4s_1(a_1^1)s_2(a_2^2) + 4s_1(a_2^1)s_2(a_1^2) + 2s_1(a_2^1)s_2(a_2^2)] \\
&= \max_{S_2}[s_1(a_1^1)s_2(a_1^2) + 4s_1(a_1^1)(1 - s_2(a_1^2)) + 4(1 - s_1(a_1^1))s_2(a_1^2) \\
&\qquad + 2(1 - s_1(a_1^1))(1 - s_2(a_1^2))] \\
&= \max_{S_2}[2 + 2s_1(a_1^1) + 2s_2(a_1^2) - 5s_1(a_1^1)s_2(a_1^2)] \qquad (13)
\end{aligned}$$

Taking the derivative with respect to $s_2(a_1^2)$ gives $2 - 5s_1(a_1^1) = 0$, or $s_1(a_1^1) = 0.4$. Therefore, the defender's minmax strategy is $(s_1(a_1^1), s_1(a_2^1)) = (0.4, 0.6)$ with a minmax value for the attacker of $\underset{S_1}{min}\ \underset{S_2}{max}\ u_2(s_1(a_i^1), s_2(a_j^2)) = 2.8$.

In a similar fashion, the mixed strategy for the attacker is found.

$$\begin{aligned}
&\underset{S_1}{max}[8s_1(a_1^1)s_2(a_1^2) + 11s_1(a_1^1)s_2(a_2^2) + 9s_1(a_2^1)s_2(a_1^2) + 7s_1(a_2^1)s_2(a_2^2)] \\
&= \underset{S_1}{max}[8s_1(a_1^1)s_2(a_1^2) + 11s_1(a_1^1)(1 - s_2(a_1^2)) + 9(1 - s_1(a_1^1))s_2(a_1^2) \\
&\qquad + 7(1 - s_1(a_1^1))(1 - s_2(a_1^2))] \\
&= \underset{S_1}{max}[-5s_1(a_1^1)s_2(a_1^2) + 4s_1(a_1^1) + 2s_2(a_1^2) + 7] \qquad (14)
\end{aligned}$$

Taking the derivative with respect to $s_1(a_1^1)$ gives $-5s_2(a_1^2) + 4 = 0$. So, $s_2(a_1^2) = 0.8$. Therefore, the attacker's minmax strategy is $(s_2(a_1^2), s_(a_2^2)) = (0.8, 0.2)$, with a minmax value for the defender of $\underset{S_2}{min}\ \underset{S_1}{max}\ u_1(s_1(a_i^1), s_2(a_j^2)) = 8.6$.

### 3.6 Extensive Form Game

Denial of service (DoS) and distributed denial of service (DDoS) are two effective and relatively simple attack types commonly deployed by attackers. These attacks are prevalent because many organizations lack proper monitoring and maintenance of their network, thereby increasing the potential for an attack. Two specific attacks that can cause a great deal of damage are the Smurf attack, which is a form of DoS attack, and reflection DDoS.

There are two defensive strategies that have proven effective in countering these attacks: disabling IP broadcasting and port blocking. Disabling IP broadcasting on network devices counters the Smurf attack by removing the attacker's ability to spoof an IP and rebroadcast over the victim's network. Port blocking reduces the usable IP space for executing either attack. This technique may not completely negate the

possibility of a reflected DDoS attack, however it can greatly reduce the attack surface [32]. The following scenario follows a extensive form game in which players utilize each of the aforementioned techniques.

An organization is evaluating defensive strategies in which to implement first, given the prevalence of the previously noted attacks. Network system administrators have decided that disabling IP broadcasting and implementing port blocking are two methods to consider. However, due to the time commitment of implementation and testing, the administrators wish to execute only one strategy immediately.

Let the players and action set be defined as before in the minmax game where the defender's actions are disabling IP broadcasting ($a_1^1$) and port blocking ($a_2^1$), and the attacker's actions are Smurf attack ($a_1^2$) and reflection DDoS ($a_2^2$). The utility function is defined as $u_2(a_j^2) = -u_1(a_i^2)$ for $i \in \delta$, $j \in \alpha$. The extensive form game tree is shown in Figure 1. Note that the utilities based on the actions taken first by the defender then the attacker are found at the terminal nodes at the base of the tree.

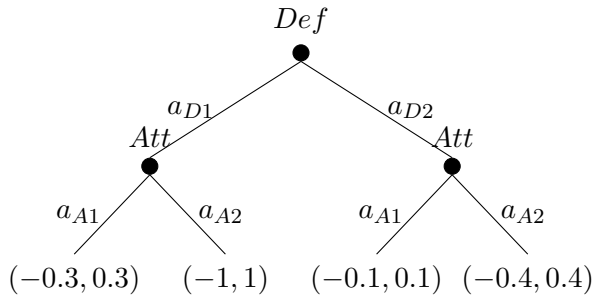



**Figure 1. Extensive Form Game Tree**

To find the Nash equilibria, the extensive form game is converted to a normal form game, called the induced normal form game [1]. The resulting game matrix is shown in Table 7.

Note that the attacker's strategies are shown as the combination of actions chosen for each branch of extensive form tree. Solving the induced normal form game gives the pure-strategy Nash equilibria $(a_2^1, (a_1^2, a_2^2))$ and $(a_2^1, (a_2^2, a_2^2))$.

**Table 7. Induced Normal Form Game Matrix**

| | | Player 2 | | | |
|---|---|---|---|---|---|
| | | $(a_1^2, a_1^2)$ | $(a_1^2, a_2^2)$ | $(a_2^2, a_1^2)$ | $(a_2^2, a_2^2)$ |
| Player 1 | $a_1^1$ | $-0.3, 0.3$ | $-0.3, 0.3$ | $-1, 1$ | $-1, 1$ |
| | $a_2^1$ | $-0.1, 0.1$ | $-0.4, 0.4$ | $-0.1, 0.1$ | $-0.4, 0.4$ |

Another type of solution, the subgame-perfect Nash equilibrium, exists for this game. The subgame-perfect Nash equilibrium is the optimal strategy played by each player found at each player decision node. Backward induction is used to find such an equilbrium, starting at the terminal nodes and progressing up, or backwards in the sequence, to the root node. Note that subgame-perfect Nash equilibria are pure-strategy Nash equilibria, but not all pure-strategy Nash equilibria are subgame-perfect Nash equilibria [21].

Using backward induction noting that the attacker plays the last action in the game, the best response of the attacker given each of the defender's actions is evaluated. Since $u_2(a_1^1, a_2^2) = 1$ and $u_2(a_2^1, a_2^2) = 0.4$, the attacker's best response to both of the defender's actions is $a_2^2$. Now considering the defender's utility between each of the two strategies, the defender receives a better payoff by choosing to play $a_2^1$. Therefore, the subgame-perfect Nash equilibria is $(a_2^1, (a_1^2, a_2^2))$ and $(a_2^1, (a_2^2, a_2^2))$ since both strategy pairs result in the path found by backward induction. In this case, the subgame-perfect Nash equilbria and the pure-strategy Nash equilibria are the same.

## 3.7 Summary

Game theory is a valuable tool in gaining insight on an array of common cyber security issues ranging from patch management to acquisitions. The applications in this section are small in scope, however the modeling techniques provide relatively simple approaches to address complex issues in the day-to-day operations of cyber security and form a foundation for the research herein.

# IV. Methodology

## 4.1 Overview

This chapter develops the components and methods used to formulate a model using game theoretical and stochastic techniques in order to evaluate the security of a cyber system. Section 4.3 provides the fundamental components that make up the game theoretical model while defining the states and actions for each player. Section 4.4 details the use of the game theoretical model within a discrete time Markov chain (DTMC). Section 4.5 leverages the previous two sections to develop a cost evaluation that provides the projected effects and risks of resource improvement allocations. Finally, Section 4.6 discusses the limitations of each component of the model.

Note that while the action space is typically a more foundational element than the state space in the development of a game, the state space is discussed first. The definition of the action space in this model is dependent on how the state space is defined, making it necessary to address the latter first. Additionally, the state space is further discussed within the context of the DTMC. While the use of the state space is similar in both scenarios, there are nuances for each that are addressed.

## 4.2 Assumptions

All data presented in this model is notional due to the lack of availability and the sensitivity of real-world data. However, the model provides a framework from which real-world data can be applied according to the needs of the system being evaluated.

All players of the game are assumed to be single units that act in unison towards a specific action. While real-world cyber security units and hostile actors may consist of a team of individuals or groups acting both in concert and independently towards a specific goal, the complexity of the internal dynamics is not captured in this model.

Additionally, players perform actions simultaneously, while the movement between games occurs sequentially and is dependent on the state in which the chain begins. Finally, each state game is independent of each other. Thus, the outcome of one state game does not affect the outcome of another.

Cyber attack and defense are represented by states with each state having Blue and Red policy options that create a unique tactical normal form game or "state game." The larger operational network security problem consists of 20 tactical normal form state games viewed as sub-problems to the operational problem. Given a state game and chosen Blue and Red policies, a transition to an adjoining state may occur with a probability dependent on the Blue and Red policy action chosen. Analysis consists of determining Red and Blue policy actions for the operational problem using game theory and the overall likelihood of termination into one of a finite set of Blue success or Red success states. The details of these games within each state is what follows.

## 4.3 Game Construction

This section defines the players of the game, the state space, and the action space for each player. These components are then used to construct the utility function and evaluate each game to find the Nash equilibria.

### Players.

The game consists of two players, Blue and Red, representing a defensive administrator unit and a hostile, or attacker, unit. While each player may consist of a team of individuals, their use is defined as a singular entity so that only one action may be taken by either player at any given time. This is often referred to as unity of effort in military operations. The set of players is defined as $N = \{1, 2\}$, where players one and two represent Blue and Red, respectively. Using notation presented in Chapter

2, $i \in N$ indicates a player within the set $N$ and $-i \in N$ indicates all other players of the game.

**State Space.**

Players perform actions that are dependent on the state they are currently in. Each state is represented by a normal form game consisting of Blue and Red allowable actions whose outcome dictates the progression through both the Blue and Red states shown in Tables 8 and 9. The Blue state space follows a notional process for detecting, identifying, and mitigating malicious activity on a network, while the Red state space follows an attack process from scanning the environment to maintaining access and hiding traces of malicious activity.

**Table 8. Blue State Space**

| State Index | State Name |
|---|---|
| 1 | Listen and Detect |
| 2 | Identify |
| 3 | Administrator Action |
| 4 | Integrity Check |

**Table 9. Red State Space**

| State Index | State Name |
|---|---|
| 1 | Scanning |
| 2 | Exploit & Elevate Privileges |
| 3 | Attack |
| 4 | Maintain Access |
| 5 | Obfuscate |

***Listen and Detect - Blue State (1).***

Before any action can be taken against an adversary (Red), the defender (Blue) must first be able to detect the adversary's presence. Therefore, this state consists of active feedback provided by defensive sensors like intrusion detection systems (IDS), intrusion prevention systems (IPS), and network health software.

***Identify - Blue State (2).***

Once hostile activity has been detected, the defensive administrators must identify what activity is taking place and, if possible, the point of origin. Information for identification comes from signature recognition devices and software like firewalls and antivirus, checking for anomalous permission changes at a root or administrative level, or changes in resource allocation that do not coincide with normal network traffic. The latter is typically noticed by surges of bandwidth utilization.

***Administrator Actions - Blue State (3).***

The certainty in identifying potentially malicious behavior informs the proper actions taken by network or system administrators. To correct this behavior, administrators can perform actions that vary greatly in severity and complexity.

Administrators may create, remove, or edit rules on the internal or external firewalls, depending on mission and network requirements. They may also perform service actions on various network devices, namely switches and routers, that adjust or repair the configuration or the ports and services that are open. Shutting down unused ports and services is often the most overlooked action and can often provide a simple solution to what may seem like a complex problem.

Finally, administrators may implement less costly actions with great effect in the form of password policies and patching. Password policies such as complexity requirements and frequent password changes are simple solutions to common password attacks, while patching provides solutions from hardware and software vendors to mitigate discovered vulnerabilities, thereby limiting the attack surface with each patch deployed.

***Integrity Check - Blue State (4).***

As a final action, administrators may perform an integrity check on hashes, directories, and files to ensure that malicious manipulation has not occurred and that no lingering software exists that may enable a hostile actor to re-engage the network.

***Scanning - Red State (1).***

The first step in an attack is to scan the target environment. This exploratory action may be performed actively by pinging devices on the victim network, passively by monitoring traffic passing into and out of the network, or by targeting users and services of the network via social engineering or interrogating open source websites hosted by the victim.

***Exploit & Elevate Privileges - Red State (2).***

Once information is obtained about the victim network, an attacker must gain access to a device and elevate privileges, preferably to a root or administrator level, so that they may gain greater mobility across the network.

***Attack - Red State (3).***

Activities that can destroy, degrade, disrupt, deny, or deceive are considered attacks. Common methods are denial of service (DoS) attacks and deploying malicious software on the victim network via file transfer, e-mail, or other means of data transfer between devices.

***Maintain Access - Red State (4).***

Oftentimes the initial attack is not the last action that a hostile actor performs.

Therefore, maintaining access to the victim network is critical to executing additional or amplifying effects to the victim.

#### *Obfuscate - Red State (5).*

Before an attacker can consider their task complete, it is important that they take action to hide traces of their activities to limit the chance of attribution or discovery. This may be accomplished by the use of hidden files and directories to store software that enables the attacker to maintain or regain access to the victim network, or by altering or deleting log files thereby removing traces of actions performed.

With the state space defined for each player, it is now defined formally within the context of the game. Let $S_i$ be the set of states of player $i$, where $i \in N$. Using the Cartesian product of the Blue and Red states, let $S$ be the complete set of states where

$$S = S_1 \times S_2 = \{(s_1, s_2) : s_1 \in S_1, s_2 \in S_2\} \tag{15}$$

### Action Space.

The possible actions for each player are defined by the respective state. These actions are exclusive to a particular player state and may not be played when the state is exited. The complete action space consists of 14 Blue actions across four states and 21 Red actions across five states.

#### *Blue Actions.*

Table 10 provides a list of Blue actions that may be played within the corresponding state. For Blue's first state, sensor data consists of feedback from an IDS, IPS, or related device, while sniffers consist of software used for monitoring network traffic.

In the second state, Blue may check the permissions of users on the network to determine if someone has unauthorized administrator privileges, check resource allocation to see if there is activity causing higher resource utilization than is typical for day-to-day operations, or check signatures detected by devices and software like firewalls and antivirus. In the third state, Blue has a number of possible actions that can

**Table 10. Blue Action Space**

| State | Index | Action |
|---|---|---|
| 1 | 1 | Sensor Data |
| | 2 | Sniffers |
| 2 | 1 | Check Permissions |
| | 2 | Resource Allocation |
| | 3 | Signatures |
| 3 | 1 | Firewall Rules |
| | 2 | Network Reconfiguration |
| | 3 | Password Policies |
| | 4 | Port/Service Management |
| | 5 | Patching |
| 4 | 1 | Check Logs |
| | 2 | Check Hashes |
| | 3 | Check Directories |
| | 4 | Check Files |

be categorized into configuration, policy, and maintenance. For configuration, Blue may change or implement firewall rules, make physical or logical changes to the network configuration, or shut down unused ports and services to limit points of entry to the network. Password policies may be implemented to counter social engineering attempts, confuse potential password cracking software used by an attacker, or mitigate unwarranted privilege escalation. Finally, patches may be deployed to improve the maintenance of the network while decreasing the potential attack surface [32].

For the final state, Blue may perform various actions that check the integrity of logs, hashes, directories, and files. Performing these actions will alert the Blue

player to malicious activity, such as rootkits and backdoors, if executed thoroughly and successfully.

***Red Actions.***

Table 9 provides a list of Red actions that can be played within the corresponding state. In the first Red state, exploratory actions are taken which consist of sending

**Table 11. Red Action Space**

| State | Index | Action |
|---|---|---|
| 1 | 1 | Pinging |
| | 2 | Channel Monitoring |
| | 3 | Traffic Monitoring |
| | 4 | Open Source |
| 2 | 1 | Trojan Horse |
| | 2 | Spoofing |
| | 3 | Obtain Credentials |
| | 4 | Inject |
| | 5 | Overflow |
| 3 | 1 | Hijacking |
| | 2 | Packet Manipulation |
| | 3 | Flood |
| | 4 | Process Manipulation |
| | 5 | Malware |
| 4 | 1 | Covert Channels |
| | 2 | Rootkit |
| | 3 | Spyware |
| | 4 | Backdoor |
| 5 | 1 | Alter Logs |
| | 2 | Hidden Directories |
| | 3 | Hidden Files |

ping requests to external victim devices in hopes of receiving a positive response, inspecting transmission frequencies typically originating from wireless sources (channel monitoring), inspecting the communication moving in and out of the network (traffic monitoring), and open source techniques like social engineering [32].

The actions in the second state leverage the knowledge gained from the first and allow for gaining access to the victim network and increasing the level of permissions so that actions within the network may occur. A common technique is using a trojan horse which masks malicious software in communication or software that seems otherwise legitimate. Additionally, any system data obtained in the previous state like IPs may be spoofed, making the attacker appear as an already established acceptable user, or the attacker may perform an overflow attack on a vulnerable service. Finally, the attacker may use intelligence gathered via open source means in order to steal passwords or hashes, or by injecting code into websites owned and managed by the victim so that administrative access can be gained [32].

Once the third stage is reached, the Red player can begin performing attacks on the victim. The complete list of potential attacks is vast, therefore this model uses five general categories of attacks. Note that these categories are not exhaustive as new attack methods are frequently discovered. One potential attack method, packet manipulation, changes the contents of the packets being received by the victim from the attacker, creating confusion on the device and forcing it to shutdown. This type of attack is part of a large category of attacks called denial of service (DoS) attacks. Another DoS attack, process manipulation, is performed in a similar way as packet manipulation, except the intended target is an active and vulnerable process like file transfer protocol (FTP) or simple network mail protocol (SNMP). A flood consists of overwhelming switching devices with IP or MAC addresses, forcing the device to allow the attacker to pass through to the network while using a illegitimate address. Finally, hijacking occurs when an attacker interrupts and steals a current active session from a legitimate user [32].

In the fourth state, Red begins to take actions that enable maintaining access to the victim's network. Backdoors may be used by an attacker to bypass security

protocols and authentication services while rootkits modify existing executable files so that root user privileges can be maintained. Taking advantage of previous actions like open source intelligence gathering, spyware may be installed on a victim device in order to monitor the activities of users, such as websites accessed, search results, and keystrokes for additional passwords and user names [32].

For the final state, the Red player masks the previous actions performed by covering their tracks throughout the victim network. This can be accomplished by altering log files or hiding directories and files that may contain software for any of the actions previously discussed [32].

From the aforementioned Blue and Red actions and states, 20 unique tactical normal form state games, denoted by $G(S)$, are created representing each state $(s_1, s_2) \in S$. Since the Blue player can be in any of the four states $\{1, 2, 3, 4\}$ while the Red player can be in any of the fives states $\{1, 2, 3, 4, 5\}$, 20 potential normal form state games are constructed. For these tactical games, let $N_{s_1}^B$ be the number of allowable actions in Blue state $s_1$ and $N_{s_2}^R$ be the number of allowable actions in Red state $s_2$. Also, let $\alpha_{(s_1,i)}^B$ be the Blue action in state $s_1$ where $i = 1, \ldots, N_{s_1}^B$ and let $\alpha_{(s_2,j)}^R$ be the Red action in state $s_2$ where $j = 1, \ldots, N_{s_2}^R$. The complete action space is defined by

$$\begin{aligned} A &= A^B \times A^R \\ &= \{(\alpha_{(s_1,i)}^B, \alpha_{(s_2,j)}^R) : s_1, s_2 \in S, i = 1, \ldots, N_{s_1}^B, j = 1, \ldots, N_{s_2}^R\} \end{aligned} \tag{16}$$

For example, $\alpha_{(1,1)}^B$ indicates the Blue action with index one (*Sensor Data*) in Blue state 1.

**Utilities.**

The utilities, or payoffs, for Blue are the probabilities of success of the played action given the corresponding Red action, defined by

$$u_B(\alpha^B_{(s_1,i)}, \alpha^R_{(s_2,j)}) = P(\alpha^B_{(s_1,i)} | \alpha^R_{(s_2,j)}) \tag{17}$$

Similarly, Red utilities are the probabilities that the played action is successful. However, each Red utility begins with a baseline probability of success, shown in Table 12 and denoted by $P(\alpha^R_{(s_2,j)})$, based on factors such as complexity and resources necessary for each action. From the baseline, the overall probability of success is

**Table 12. Red Baseline Probabilities of Success**

| Action | Baseline Probability |
|---|---|
| Pinging | 0.7 |
| Channel Monitoring | 0.3 |
| Traffic Monitoring | 0.5 |
| Open Source | 0.4 |
| Trojan Horse | 0.5 |
| Spoofing | 0.3 |
| Obtain Credentials | 0.4 |
| Inject | 0.2 |
| Overflow | 0.5 |
| Hijacking | 0.3 |
| Packet Manipulation | 0.2 |
| Flood | 0.3 |
| Process Manipulation | 0.2 |
| Malware | 0.4 |
| Covert Channels | 0.1 |
| Rootkit | 0.3 |
| Spyware | 0.5 |
| Backdoor | 0.3 |
| Alter Logs | 0.2 |
| Hidden Directories | 0.5 |
| Hidden Files | 0.6 |

determined by the product of the baseline probability and the probability that the corresponding Blue action fails. Thus, the Red utility function is defined as

$$u_R(\alpha^B_{(s_1,i)}, \alpha^R_{(s_2,j)}) = P(\alpha^R_{(s_2,j)})(1 - u_B(\alpha^B_{(s_1,i)}, \alpha^R_{(s_2,j)})) \tag{18}$$

Each game is constructed as shown in Appendix A. Note that each payoff matrix reads with the Blue actions down the rows, Red actions across the columns, and payoffs ordered by Blue then Red. In each game, the pure strategy Nash equilibria (PSNE) and mixed strategy Nash equilibria (MSNE) are found to determine which strategies are dominant as well as the probability that each strategy is played. The PSNE and MSNE for each game are presented in Appendix B.

### 4.4 Discrete Time Markov Chain.

In this section, the previously constructed tactical level games are used to construct a DTMC, forming the larger operational network security problem. To establish a winner of the stochastic game, states are added for both Blue and Red that are set as "win" states, increasing the number of Blue states to five and Red to six. These additional states are $\{(5, s_2) : s_2 \in S_2 \backslash \{6\}\}$ for a Blue win, $\{(s_1, 6) : s_1 \in S_1 \backslash \{5\}\}$ for a Red win, and $\{(5, 6)\}$ for a stalemate between the players. From the perspective of the Blue player, state $(5, 6)$ is not an ideal end state since Red can still accomplish their mission even though Blue has stopped any follow-up action, however it is preferable to any of the Red win states.

#### State Transitions.

Movement between states occurs by either a forward motion or a self-transition, however neither player can return to a previously visited state. For example, if the game is currently in state $(1, 2)$, a transition can be made to $(1, 2)$, $(1, 3)$, $(2, 2)$, or

$(2,3)$, but will never transition to $(1,1)$. Given this rule, the transition matrix is constructed.

Let $\boldsymbol{\lambda}_s$ be a matrix of the tactical game MSNE in state $s = (s_1, s_2) \in S$ of size $|A^B_{s_1}| \times |A^R_{s_2}|$ where $A^B_{s_1} \in A^B$ is Blue's action space for state $s_1 \in S_1$, $A^R_{s_2} \in A^R$ is Red's action space for state $s_2 \in S_2$, and each entry of $\boldsymbol{\lambda}_s$ is the product of mixed strategy probabilities for each Blue and Red action pairing in state $s$. Let $\boldsymbol{\Lambda}$ be a matrix of the operational network security problem MSNE of size $|A^B| \times |A^R|$ constructed of all $\boldsymbol{\lambda}_s$ as shown below.

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\lambda}_{(1,1)} & \boldsymbol{\lambda}_{(1,2)} & \cdots & \boldsymbol{\lambda}_{(1,|S_2|)} \\ \boldsymbol{\lambda}_{(2,1)} & \boldsymbol{\lambda}_{(2,2)} & \cdots & \boldsymbol{\lambda}_{(2,|S_2|)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\lambda}_{(|S_1|,1)} & \boldsymbol{\lambda}_{(|S_1|,1)} & \cdots & \boldsymbol{\lambda}_{(|S_1|,|S_2|)} \end{pmatrix} \tag{19}$$

Let $\boldsymbol{\delta}^B_s$ be the strategic form utility matrix of Blue in state $s = (s_1, s_2) \in S$ and $\boldsymbol{\delta}^R_s$ be the strategy form utility matrix of Red, each utility matrix of equal size as $\boldsymbol{\lambda}_s$. Furthermore, let $\boldsymbol{\Delta}^B$ be the complete operational problem strategic form utility matrix for Blue consisting of $\boldsymbol{\delta}^B_s$ and $\boldsymbol{\Delta}^R$ be the complete operational problem strategic form utility matrix for Red consisting of $\boldsymbol{\delta}^R_s$. Both $\boldsymbol{\Delta}^B$ and $\boldsymbol{\Delta}^R$ are of equal size as $\boldsymbol{\Lambda}$.

$$\boldsymbol{\Delta}^B = \begin{pmatrix} \boldsymbol{\delta}^B_{(1,1)} & \boldsymbol{\delta}^B_{(1,2)} & \cdots & \boldsymbol{\delta}^B_{(1,|S_2|)} \\ \boldsymbol{\delta}^B_{(2,1)} & \boldsymbol{\delta}^B_{(2,2)} & \cdots & \boldsymbol{\delta}^B_{(2,|S_2|)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\delta}^B_{(|S_1|,1)} & \boldsymbol{\delta}^B_{(|S_1|,1)} & \cdots & \boldsymbol{\delta}^B_{(|S_1|,|S_2|)} \end{pmatrix} \tag{20}$$

$$\boldsymbol{\Delta}^R = \begin{pmatrix} \boldsymbol{\delta}^R_{(1,1)} & \boldsymbol{\delta}^R_{(1,2)} & \cdots & \boldsymbol{\delta}^R_{(1,|S_2|)} \\ \boldsymbol{\delta}^R_{(2,1)} & \boldsymbol{\delta}^R_{(1,2)} & \cdots & \boldsymbol{\delta}^R_{(2,|S_2|)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\delta}^R_{(|S_1|,1)} & \boldsymbol{\delta}^R_{(|S_1|,2)} & \cdots & \boldsymbol{\delta}^R_{(|S_1|,|S_2|)} \end{pmatrix} \tag{21}$$

For those state games in which multiple MSNE exist, only the first listed MSNE is used for the DTMC. Since each of the MSNE result in equal expected utilities for each player, generality is not lost with this choice.

The probability of success for the Blue action in state $k$ per the MSNE is denoted by $x_k$, while the probability of success for the Red action is denoted by $y_k$. Then the Blue and Red success probabilities, respectively, with matrices indexed on $(a, b)$ are calculated by

$$x_k = \sum_{a \in |A^B_{s_1}|} \sum_{b \in |A^R_{s_2}|} \boldsymbol{\lambda}_k(a,b) \boldsymbol{\delta}^B_k(a,b) \tag{22}$$

$$y_k = \sum_{a \in |A^B_{s_1}|} \sum_{b \in |A^R_{s_2}|} \boldsymbol{\lambda}_k(a,b) \boldsymbol{\delta}^R_k(a,b) \tag{23}$$

where $\mathbf{x} = \{x_k : k \in S\}$ and $\mathbf{y} = \{y_k : k \in S\}$.

From Equations 22 and 23, the transition matrix $\mathbf{P}$ is constructed. Transitions between states are denoted in the form of $P_{j,k}$, where $j$ is the exited state and $k$ is the state being entered [27]. The transition probabilities are calculated by

$$P_{(s_1,s_2),(s_1+1,s_2)} = x_{(s_1,s_2)}(1 - y_{(s_1,s_2)}) \tag{24}$$

$$P_{(s_1,s_2),(s_1,s_2+1)} = (1 - x_{(s_1,s_2)})y_{(s_1,s_2)} \tag{25}$$

$$P_{(s_1,s_2),(s_1+1,s_2+1)} = x_{(s_1,s_2)}y_{(s_1,s_2)} \tag{26}$$

$$P_{(s_1,s_2),(s_1,s_2)} = 1 - (P_{(s_1,s_2),(s_1+1,s_2)} + P_{(s_1,s_2),(s_1,s_2+1)} + P_{(s_1,s_2),(s_1+1,s_2+1)}) \tag{27}$$

Note that for Equation 27, since win states are absorbing states, $P_{(s_1,s_2),(s_1,s_2)} = 1$ for each win state. The transition matrix partitions are shown in Appendix D due to the size of the matrix, with the exception of $\mathbf{P}_5 = \mathbf{I}_5$. The transition matrix consisting of partitions is shown below. While not explicitly shown below, overlap of columns occurs between partitions. Thus, Equation 28 is not exactly to scale.

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_1 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{P}_2 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{P}_3 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{P}_4 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{P}_5 \end{pmatrix} \tag{28}$$

**Stationary Probabilities.**

Using techniques presented in Section 2.6, the stationary probabilities are found. Let $\boldsymbol{\pi} = \{\pi_k : k \in S\}$ be the stationary probability vector, and $T$ the set of time steps of the DTMC such that $T = \{1, 2, \ldots, t\}$. Since the Markov chain is ergodic, then $\lim_{t\to\infty} \mathbf{P}^t_{j,k}$ converges to a non-negative unique stationary distribution and is indifferent to the initial state of the Markov chain. [27]. The convergence of the limiting distribution can be seen by noting the point in which the entries of the matrix $\mathbf{P}^t_{j,k}$

stabilizes or via graphical representation of each iteration. Alternatively, the stationary distribution may be found by solving the system of linear equations consisting of $\boldsymbol{\pi}\mathbf{P}$ and $\sum_{k\in S}\pi_k = 1$. For this research, the stationary probabilities are found via the limiting probabilities method.

Once the stationary probabilities are found, an evaluation is made to determine the probability of winning for each player based on the stationary probabilities for their respective win states. Note that $\sum_{k\in S}\pi_k = 1$.

**Transient State Analysis.**

Removing the absorbing states from $\mathbf{P}$ results in the transient state matrix defined by $\mathbf{Q}$, shown below. Let $S' = \{(s_1, s_2) : s_1 \in S_1', s_2 \in S_2'\}$ be the set of transient states. The partitions of $\mathbf{Q}$ can be found in Appendix F.

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & 0 & 0 & 0 \\ 0 & \mathbf{Q}_2 & 0 & 0 \\ 0 & 0 & \mathbf{Q}_3 & 0 \\ 0 & 0 & 0 & \mathbf{Q}_4 \end{pmatrix} \tag{29}$$

From $\mathbf{Q}$, the expected time in state $k$ given the Markov chain begins in state $j$ is derived, denoted by $m_{j,k}$. The matrix $\mathbf{M}$ consists of each $m_{j,k}$ for $j, k \in S'$ [27]. Applying the method presented in Chapter 2,

$$\mathbf{M} = (\mathbf{I} - \mathbf{Q})^{-1} = \begin{pmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} \\ 0 & \mathbf{M}_{3,2} \\ 0 & \mathbf{M}_{4,2} \end{pmatrix} \tag{30}$$

where $\mathbf{I}$ is the identity matrix of equal size to $\mathbf{Q}$. The partitions of $\mathbf{M}$ are found in Appendix G.

The final portion of the transient state analysis is finding the probability that state $k$ is entered given the Markov chain begins in state $j$, whose matrix is denoted by $\mathbf{F}$. For each entry $f_{j,k} \in \mathbf{F}$ and $j, k \in S'$, $f_{j,k}$ is the probability of transitioning to state $k$ given the Markov chain begins in $j$ determined by

$$f_{j,k} = \frac{m_{j,k}}{m_{k,k}} \tag{31}$$

Given Equation 31 [27], $\mathbf{F}$ is constructed from the appropriate entries found in $\mathbf{M}$, whose values can be used to evaluate the likelihood of paths and outcomes in the stochastic game.

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{1,1} & \mathbf{F}_{1,2} \\ \mathbf{F}_{2,1} & \mathbf{F}_{2,2} \\ 0 & \mathbf{F}_{3,2} \\ 0 & \mathbf{F}_{4,2} \end{pmatrix} \tag{32}$$

### 4.5 Cost Evaluation

With the components of the DTMC found, an evaluation is made for the cost incurred by Blue given the cost of the damages caused by the actions of Red and the probability of success of actions taken by Blue. A cost evaluation is made using DTMC and optimization, the former considering the expected cost of damage in each state and the latter for evaluating the cost of repairing or upgrading components corresponding with Blue actions while attempting to minimize the cost of damage by increasing the probability of success for Blue actions.

**Evaluation via DTMC.**

A cost evaluation based on the DTMC is made by utilizing the results of the mean time in each state, $\mathbf{M}$, and the expected damage cost to Blue's system given actions taken by Red.

Let $\mathbf{d} = \begin{pmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \cdots & \mathbf{d}_{|S_2'|} \end{pmatrix}$ be an expected baseline damage cost vector from actions taken by Red, where $d_{s_2,j} \in \mathbf{d}_{s_2}$ is the baseline damage cost for action $j$ in Red state $s_2 \in S_2'$. The baseline cost does not consider the effectiveness of Blue's response nor the MSNE, however it is used to find the estimated cost of damage. The baseline damage costs are shown in Table 13. Note that these values are notional.

**Table 13. Red Baseline Damage Cost**

| **Action** | **Baseline Damage Cost (in thousands)** |
|---|---|
| Pinging | 1 |
| Channel Monitoring | 2 |
| Traffic Monitoring | 2 |
| Open Source | 5 |
| Trojan Horse | 4 |
| Spoofing | 3 |
| Obtain Credentials | 3 |
| Inject | 3.5 |
| Overflow | 5 |
| Hijacking | 6 |
| Packet Manipulation | 5 |
| Flood | 8 |
| Process Manipulation | 5 |
| Malware | 3 |
| Covert Channels | 2.5 |
| Rootkit | 5 |
| Spyware | 3 |
| Backdoor | 3 |
| Alter Logs | 2 |
| Hidden Directories | 1.5 |
| Hidden Files | 1.5 |

Let $c_s$ be the estimated cost of damage in thousands of dollars incurred by Blue given the baseline cost of Red's action taken in state $s \in S'$, and

$$\mathbf{c} = \begin{pmatrix} c_{(1,1)} & c_{(1,2)} & \cdots & c_{(5,6)} \end{pmatrix}^T \tag{33}$$

be the vector of estimated damage cost of all states. Since the absorbing states are designated as "win" states, no damage cost is assigned and is not considered in the cost evaluation. The estimated cost of damage for $s = (s_1, s_2) \in S'$ is defined by

$$c_s = \sum_{a \in |A^B_{s_1}|} \sum_{b \in |A^R_{s_2}|} (1 - \boldsymbol{\delta}^B_s(a,b))\, \boldsymbol{\lambda}_s(a,b)\, \mathbf{d}_{s_2}(b) \tag{34}$$

Taking the product of the estimated cost of damage and the matrix of mean times in each state gives the matrix of the total expected cost of damage in each state, denoted by $\mathbf{C}$. Equation 35 provides the method in which each entry is found and Equation 36 shows the matrix of partitions of the expected cost per state. Each partition is shown in Appendix I.

$$\mathbf{C}(a,b) = \mathbf{M}(a,b)\mathbf{c}(a), \quad \forall a \in |A^B|, \forall b \in |A^R| \tag{35}$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \\ 0 & \mathbf{C}_{3,2} \\ 0 & \mathbf{C}_{4,2} \end{pmatrix} \tag{36}$$

**Evaluation via Mathematical Programming.**

With the expected cost of damage determined utilizing the DTMC model, a natural extension is assessing how variability in network devices may affect the original DTMC model and the expected damage cost. Variability for this formulation considers a scenario in which components that affect the actions taken by Blue can either be repaired or upgraded given a threshold of allowable funding. Components of systems become degraded over time, either through day-to-day use or malicious activity. Therefore, a decision must inevitably be made as to what action must be taken for the component.

To model a scenario of this kind, an integer program (IP) model is constructed. If a component is repaired, it is assumed that the component is restored to full functionality so that the corresponding probability of success for the Blue action is unaffected. If a component is upgraded, the corresponding probability of success is changed to that of the new component. These probabilities must be enumerated but need not be better than before and are the expected probability given the upgrade. An upgrade may be a full replacement of the component or a change of the existing component that affects the performance, for example installing additional storage capacity. If neither option is viable due to lack of funds, the component in which no action is taken will suffer from degradation, resulting in a reduction from the original utilities.

The objective in this scenario is to maximize the probability of success for each component given a designated resource improvement allocation cost threshold. Let $x_a$, $y_a$, and $z_a$ be binary decision variables for the decision to repair, upgrade, or no action, respectively, for $a \in |A^B|$. Also, let $c_a^r$ and $c_a^u$ be the flat cost of repairing or upgrading the component before considering the expected damage cost, respectively. Similarly, $r_a$, $u_a$, and $n_a$ are defined as the total cost of repair, upgrade, and no

action, respectively, considering both the flat cost and the expected damage cost. In the case of the cost of no action, no flat cost is assigned and the value is based only on the expected damage cost given the level of effectiveness for each component, $\boldsymbol{\mu}$. The level of effectiveness is defined as the percent of the initial utility that can be expected from each component once degraded. Let $T$ denote the designated cost threshold for resource improvement allocations that must not be exceeded. The matrix of Blue utilities across the operational problem for the decision to repair or upgrade are defined as $\boldsymbol{\Delta}_r^B$ and $\boldsymbol{\Delta}_u^B$, respectively. Finally, $\boldsymbol{\Lambda}$ is as defined previously in Equation 19 and $\mathbf{d}$ is as defined on page 60.

$$\text{Maximize} \quad \sum_{a\in|A^B|}\sum_{b\in|A^R|} \boldsymbol{\Delta}_r^B(a,b)(x_a+\boldsymbol{\mu}(a)z_a)+\boldsymbol{\Delta}_u^B(a,b)y_a \tag{37a}$$

$$\text{s.t.} \quad r_a = c_a^r + \sum_{b\in A^R}(1-\boldsymbol{\Delta}_r^B(a,b))\boldsymbol{\Lambda}(a,b)\mathbf{d}(b), \qquad \forall a\in|A^B| \tag{37b}$$

$$u_a = c_a^u + \sum_{b\in A^R}(1-\boldsymbol{\Delta}_u^B(a,b))\boldsymbol{\Lambda}(a,b)\mathbf{d}(b), \qquad \forall a\in|A^B| \tag{37c}$$

$$n_a = \sum_{b\in A^R}(1-\boldsymbol{\mu}(a)\boldsymbol{\Delta}_r^B(a,b))\boldsymbol{\Lambda}(a,b)\mathbf{d}(b), \qquad \forall a\in|A^B| \tag{37d}$$

$$r_a x_a + u_a y_a + n_a z_a \leq T, \qquad \forall a\in|A^B| \tag{37e}$$

$$x_a + y_a + z_a = 1, \qquad \forall a\in|A^B| \tag{37f}$$

$$0\leq \boldsymbol{\Delta}_r^B(a,b)\leq 1, \qquad \forall a\in|A^B|,\forall b\in|A^R| \tag{37g}$$

$$0\leq \boldsymbol{\Delta}_u^B(a,b)\leq 1, \qquad \forall a\in|A^B|,\forall b\in|A^R| \tag{37h}$$

$$0\leq \boldsymbol{\mu}(a)\leq 1 \qquad \forall a\in|A^B| \tag{37i}$$

$$x_a, y_a, z_a \in \{0,1\}, \qquad \forall a\in|A^B| \tag{37j}$$

$$c_a^r, c_a^u, r_a, u_a, n_a, T \geq 0, \qquad \forall a\in|A^B|,\forall b\in|A^R| \tag{37k}$$

Equations 37b, 37c and 37d define the total cost of repair, upgrade, and no action, respectively. Since this model requires that only one of the decisions to repair, upgrade, or no action be made for each component, Equation 37j restricts the sum of the

binary variables to be one, forcing only one of the decisions to occur. Additionally, the total cost of each decision bounded by the desired cost threshold is expressed in Equation 37e. Finally, each $\boldsymbol{\Delta}_r^B(a,b)$ and $\boldsymbol{\Delta}_u^B(a,b)$ is a probability of Blue success and therefore must be a real number between zero and one. Similarly, $\boldsymbol{\mu}(a)$ is restricted to be a real number value between zero and one. All other variables must be at least zero as shown in the final constraint.

## 4.6 Limitations

First, the DTMC does not consider any backwards transitions to preceding states. A backwards transition is defined as any state transition in which a player returns to a previously visited state. While including these transitions would bring more fidelity and real-world accuracy to the model, it also increases the complexity of the problem. If backwards transitions are desired, the cause and effect of the transition must be considered.

Additionally, this model does not consider the rate in which each Blue and Red action occurs. These rates would make it possible for a player to perform multiple actions before the other player could perform a single action. Instead, Red actions and Blue actions are assumed to be aggregated into a single uniform time step. Furthermore, time steps are treated as equal and not well defined since the rates of actions are not defined.

## 4.7 Summary

This chapter provides a detailed overview on the methods and techniques used to address the research objective posed in Section 1.3. First, underlying assumptions are addressed so that the reasoning behind each technique is understood. Next, the normal form games are constructed and defined with the players, action space, state

space, and utility functions, while also defining the terms used in each of those game components.

Using the structure and solutions from the normal form games, a DTMC model is defined and presented. First, states are defined from the normal form games and the formulation of each state transition is shown. Next, stationary probabilities are found so that the probability of a win for each player can be determined. Continuing from the transition matrix, transient state analysis can be used to determine the expected time in each state and the probability of reaching each state given the starting state of the Markov chain. The expected time in state values are further leveraged to provide a cost evaluation on the expected damage cost per state given the expected time spent in that state. Finally, an IP is defined so that a determination may be made should repairs and upgrades to components for each action be required or desired.

# V. Analysis

## 5.1 Overview

Analysis provides insight into the structure of the game, ramifications of policy choices, sensitivity to information and modifications, and impacts of resource allocation decisions. Section 5.2 begins with the construction and results of the 20 state games with each player using randomized or mixed strategies. Sensitivity analysis for each state game is discussed to identify the bounds of Blue utilities in which the Nash equilibria are maintained. The calculations for player utilities and Nash equilibria are performed with manual calculations, the open-source software GAMBIT [33], and MATLAB. See Appendix J and K for the MATLAB code used.

Section 5.3 discusses the results of the application of the state game outcomes within a DTMC. Stationary probabilities are found to determine the probability of a Blue win, Red win, and stalemate. Transient analysis is then performed to determine the most likely state progression of the stochastic game, appropriate policies for each player, and an expected cost of damage to the Blue player's network. This section also explores how the results of the DTMC change when a player decides to play a sub-optimal strategy in at least one of the state games. All calculations for the DTMC are performed with MATLAB. The code can be viewed in Appendix J and K.

Finally, Section 5.4 provides the results of determining an expected cost of damage to the Blue player's network and a cost evaluation should the Blue player decide to repair, upgrade, or take no action for components affecting the playable strategies. This evaluation demonstrates the affects and risk involved in determining the allocation of funds for resource improvement. The mathematical programming calculations

are performed using the commercial solver LINGO. The code used is shown in Appendix L.

In practice, subject matter expert (SME) experience or real-time data acquired from an isolated test network environment is ideal to provide the greatest fidelity to the capabilities of the network of interest. However, such resources are not available at the time of writing, so nominal values are used. The efficacy of the model remains unaffected since the focus of the research is on how data is evaluated by the model, regardless of the data source.

## 5.2 State Game Analysis

The evaluation of the state games is performed based on the existence and number of pure-strategy Nash equilibria (PSNE). While the calculations for each game are conducted using the same techniques throughout, those games in which a unique PSNE occurs require fewer calculations and can often be solved by inspection. Considering this case first, the state game $G(1,1)$ is used as an example, shown in Table 14.

**Table 14. State Game G(1,1)**

| | | Red | | | |
|---|---|---|---|---|---|
| | | $\alpha^R_{(1,1)}$ | $\alpha^R_{(1,2)}$ | $\alpha^R_{(1,3)}$ | $\alpha^R_{(1,4)}$ |
| Blue | $\alpha^B_{(1,1)}$ | $0.2, 0.56$ | $0.1, 0.27$ | $0.2, 0.4$ | $0.1, 0.36$ |
| | $\alpha^B_{(1,2)}$ | $0.1, 0.63$ | $0.2, 0.24$ | $0.3, 0.35$ | $0.05, 0.38$ |

By first evaluating the payoffs for each of Red's strategies, the utilities for Red's strategy $\alpha^R_{(1,1)}$ are strictly greater than the utilities than any other utility should Red choose another strategy given any Blue actions. Thus, all of Red's strategies with the

exception of the dominant strategy $\alpha^R_{(1,1)}$ are removed. The state game is now reduced to the first column as shown in Table 15.

**Table 15. Reduced State Game G(1,1)**

| | | Red |
|---|---|---|
| | | $\alpha^R_{(1,1)}$ |
| Blue | $\alpha^B_{(1,1)}$ | $0.2, 0.56$ |
| | $\alpha^B_{(1,2)}$ | $0.1, 0.63$ |

In the reduced game, Blue's strategy $\alpha^B_{(1,1)}$ strictly dominates $\alpha^B_{(1,2)}$. Thus, Blue's second strategy is removed and the PSNE $(\alpha^B_{(1,1)}, \alpha^R_{(1,1)})$ is found.

For each state game in which there are multiple PSNE or no PSNE exist, the feasibility of removing dominated strategies is evaluated first using iterated removal of dominant strategies. For those games in which dominated strategies existed, the game is reduced accordingly. Regardless of the existence of dominated strategies, the mixed-strategy Nash equilibria (MSNE) are found since they are guaranteed to exist, even in the event that no PSNE exist. To demonstrate the calculations, consider the state game $G(4, 5)$ shown in Table 16.

**Table 16. State Game G(4,5)**

| | | Red | | |
|---|---|---|---|---|
| | | $\alpha^R_{(5,1)}$ | $\alpha^R_{(5,2)}$ | $\alpha^R_{(5,3)}$ |
| Blue | $\alpha^B_{(4,1)}$ | $0.5, 0.1$ | $0.3, 0.35$ | $0.3, 0.42$ |
| | $\alpha^B_{(4,2)}$ | $0.1, 0.18$ | $0.1, 0.45$ | $0.1, 0.54$ |
| | $\alpha^B_{(4,3)}$ | $0.3, 0.14$ | $0.7, 0.15$ | $0.1, 0.54$ |
| | $\alpha^B_{(4,4)}$ | $0.1, 0.18$ | $0.2, 0.4$ | $0.6, 0.24$ |

Observing Red's strategies, it is seen that $\alpha^R_{(5,1)}$ is dominated by both $\alpha^R_{(5,2)}$ and $\alpha^R_{(5,3)}$, therefore it is removed. From the reduced state game, Blue's strategies are

observed and it is seen that $\alpha^B_{(4,2)}$ is strictly dominated by both $\alpha^B_{(4,1)}$ and $\alpha^B_{(4,4)}$, and weakly dominated by $\alpha^B_{(4,3)}$. Therefore, $\alpha^B_{(4,2)}$ is removed. The reduced state game is shown in Table 17 with no remaining dominated pure strategies.

**Table 17. Reduced State Game G(4,5)**

| | | Red | |
|---|---|---|---|
| | | $\alpha^R_{(5,2)}$ | $\alpha^R_{(5,3)}$ |
| Blue | $\alpha^B_{(4,1)}$ | $0.3, 0.35$ | $0.3, 0.42$ |
| | $\alpha^B_{(4,3)}$ | $0.7, 0.15$ | $0.1, 0.54$ |
| | $\alpha^B_{(4,4)}$ | $0.2, 0.4$ | $0.6, 0.24$ |

The game is now ready to be evaluated for PSNE. Taking the Blue player's perspective, it is expected that Red will play $\alpha^R_{(5,3)}$ since the greatest utility can be gained with $u_R(\alpha^B_{(4,3)}, \alpha^R_{(5,3)}) = 0.54$. Therefore, Blue's best response is $BR_B(\alpha^R_{(5,3)}) = \alpha^B_{(4,4)}$ where $u_B(\alpha^B_{(4,4)}, \alpha^R_{(5,3)}) = 0.6$. Checking Red's strategy decision again, it is observed that Red's utility can be improved with $u_R(\alpha^B_{(4,4)}, \alpha^R_{(5,2)}) = 0.4$. This trend continues with each iteration, indicating that no PSNE exists for the Blue player. Evaluating the reduced game from the Red player's perspective results in no existing PSNE. Therefore, the MSNE must be identified.

The reduced state game is solved first by finding the mixed strategy for Blue by making Red indifferent to their actions. Let $q_1$ be the probability of Blue playing $\alpha^B_{(4,1)}$, $q_2$ be the probability of $\alpha^B_{(4,3)}$, and $1 - q_1 - q_2$ be the probability of $\alpha^B_{(4,4)}$. Since Red is indifferent in the strategy they play, $Eu_R(\alpha^R_{(5,2)}) = Eu_R(\alpha^R_{(5,3)})$. Blue's mixed strategy is now found in the following manner:

$$Eu_R(\alpha^R_{(5,2)}) = Eu_R(\alpha^R_{(5,3)})$$

$$0.35q_1 + 0.15q_2 + 0.4(1 - q_1 - q_2) = 0.42q_1 + 0.54q_2 + 0.24(1 - q_1 - q_2)$$

$$q_1 = 0.696 - 2.391q_2 \tag{38}$$

Substituting for $q_1$ shows that $q_2 < 0$, contradicting $0 \leq q_2 \leq 1$. Therefore, it must be true that one of Blue's strategies will never be played. By checking for a dominated strategy by mixed strategy, let $q_1 = 0$ and assume that each of Blue's other two strategies are played with equal probability. Solving, it is found that $Eu_B(\alpha^R_{(5,2)}) = 0.5(0.7) + 0.5(0.2) = 0.45$ and $Eu_B(\alpha^R_{(5,3)}) = 0.5(0.1) + 0.5(0.6) = 0.35$. Since these expected utilities are each greater than $u_B(\alpha^B_{(4,1)}, \alpha^R_{(5,2)})$ and $u_B(\alpha^B_{(4,1)}, \alpha^R_{(5,3)})$, Blue will never play $\alpha^B_{(4,1)}$ and can be removed. Reevaluating the further reduced state game shows that $q_2 = 0.29$ and $1 - q_2 = 0.71$. Thus, Blue's mixed strategy is $(0, 0, 0.29, 0.71)$. Solving Red's mixed strategy in a similar manner results in a mixed strategy of $(0, 0.5, 0.5)$. Therefore, the MSNE for $G(4, 5)$ is $\{(0, 0, 0.29, 0.71), (0, 0.5, 0.5)\}$.

Using the above techniques, the PSNE and MSNE are found for all 20 state games. The results of the calculations are found in Appendix B.

### Stability of Nash Equilibria

Sensitivity analysis of the Nash equilibria can be performed to determine the stability of the equilibria for each state game. Since Blue's utility directly affects Red's utility, the sensitivity of the equilibria is determined by the interval of $u_B(\alpha^B_{(s_1,i)}, \alpha^R_{(s_2,j)})$ by which the equilibria holds. To demonstrate how this is executed, sensitivity analysis on the equilibrium for $G(1, 1)$ is evaluated (see Table 14 on page 67 for the game matrix).

First, recall that for the PSNE of $G(1,1)$ Blue's utility is $u_B(\alpha^B_{(1,1)}, \alpha^R_{(1,1)}) = 0.2$. In order for the PSNE to remain the same it must remain a dominant strategy for both Blue and Red. Noting the removal of dominant strategies as previously performed, this means that $u_B(\alpha^B_{(1,1)}, \alpha^R_{(1,1)}) > 0.1$ or $u_R(\alpha^B_{(1,1)}, \alpha^R_{(1,1)})$ must remain greater than the next best Red utility given Blue plays $\alpha^B_{(1,1)}$. Thus, $u_R(\alpha^B_{(1,1)}, \alpha^R_{(1,1)}) > 0.4$.

Recalling the utility function for Red defined in Section 4.3, the following must hold true:

$$P(\alpha^R_{(1,1)})(1 - u_B(\alpha^B_{(1,1)}, \alpha^R_{(1,1)}) > 0.4 \tag{39}$$

Substituting $P(\alpha^R_{(1,1)})$ for the corresponding value in Section 4.3 and solving for $u_B(\alpha^B_{(1,1)})$,

$$\begin{aligned} 0.7(1 - u_B(\alpha^B_{(1,1)}, \alpha^R_{(1,1)}) &> 0.4 \\ u_B(\alpha^B_{(1,1)}) &< 0.4286 \end{aligned} \tag{40}$$

Therefore, the PSNE for $G(1,1)$ is maintained as long as $0.1 < u_B(\alpha^B_{(1,1)}, \alpha^R_{(1,1)}) < 0.4286$ or $0.1 < u_B(\alpha^B_{(1,2)}, \alpha^R_{(1,1)}) < 0.4571$. Note that the lower bound on the equilibrium Blue utility is formed by a comparison with the Blue utility across other Blue strategies while the upper bound is formed by a comparison with the Red utility across other Red strategies.

To show how this method applies to a state game in which no PSNE exists, recall $G(4,5)$ previously shown in Table 16. The sensitivity for the Nash equilibria $(\alpha^B_{(4,3)}, \alpha^R_{(5,2)})$ and $(\alpha^B_{(4,3)}, \alpha^R_{(5,3)})$ are shown while the remaining two MSNE action pairs can be evaluated in the same manner. Proceeding as before, the Blue utilities are evaluated first so that a lower bound may be established, followed by the Red utilities.

For the equilibrium $(\alpha^B_{(4,3)}, \alpha^R_{(5,2)})$, observing the utilities across the Blue strategies given Red plays $(\alpha^R_{(5,2)})$ shows that $u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,2)}) > 0.3$. For $(\alpha^B_{(4,3)}, \alpha^R_{(5,3)})$, an interesting case occurs when the Blue utility for this equilibrium is not strictly greater than the next best Blue utility given Red plays $\alpha^R_{(5,2)}$. In this case, decreasing the Blue utility does not affect the equilibrium since doing so increases the corresponding Red utility and does not assist in dominance among Blue strategies. Thus, $u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,3)}) > 0$.

Evaluating the sensitivity of the Red utilities, in order to maintain the equilibria given Blue plays $\alpha^B_{(4,3)}$, the following must be true:

$$P(\alpha^R_{(5,2)})(1 - u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,2)}) > 0.14 \tag{41a}$$

$$P(\alpha^R_{(5,3)})(1 - u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,3)}) > 0.15 \tag{41b}$$

Substituting for $\alpha^R_{(5,2)}$ and $\alpha^R_{(5,3)}$,

$$\begin{aligned} 0.5(1 - u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,2)}) &> 0.14 \\ u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,2)}) &< 0.72 \qquad (42a) \\ 0.6(1 - u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,3)}) &> 0.15 \\ u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,3)}) &< 0.75 \qquad (42b) \end{aligned}$$

Therefore, $(\alpha^B_{(4,3)}, \alpha^R_{(5,2)})$ remains an equilibrium for $0.4 < u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,3)}) < 0.75$, and $(\alpha^B_{(4,3)}, \alpha^R_{(5,2)})$ for $0 < u_B(\alpha^B_{(4,3)}, \alpha^R_{(5,3)}) < 0.75$.

Using this technique, the stability of the Nash equilibria for each state can be evaluated by considering thresholds of strategy domination across Blue and Red strategies.

## 5.3 DTMC Analysis

Using the previously found Nash equilibria for each tactical normal form game, a DTMC is constructed. Mixed strategies shown in Appendix B are used as weights on the utilities associated with the Nash equilibria in order to determine the probability of Blue success and Red success in each state game.

To demonstrate how the determination of the probability of success for each player leverages the MSNE as weights, the state game $G(4,5)$ is used with MSNE of $\{(0,0,0.29,0.71),(0,0.5,0.5)\}$. The associated utility values are provided in Appendix A. Using Equations 22 and 23 found in Section 4.4,

$$\begin{aligned} x_{(4,5)} &= 0.29(0.5)(0.7) + 0.29(0.5)(0.1) + 0.71(0.5)(0.2) + 0.71(0.5)(0.6) \\ &= 0.4 \end{aligned} \tag{43a}$$

$$\begin{aligned} y_{(4,5)} &= 0.29(0.5)(0.15) + 0.29(0.5)(0.54) + 0.71(0.5)(0.4) + 0.71(0.5)(0.24) \\ &= 0.3273 \end{aligned} \tag{43b}$$

Substituting these values into Equations 24 through 27 in Section 5.3,

$$P_{(4,5),(4,6)} = (1 - x_{(4,5)})y_{(4,5)} = 0.1964 \tag{44a}$$

$$P_{(4,5),(5,5)} = x_{(4,5)}(1 - y_{(4,5)}) = 0.2691 \tag{44b}$$

$$P_{(4,5),(5,6)} = x_{(4,5)}y_{(4,5)} = 0.1309 \tag{44c}$$

$$P_{(4,5),(4,5)} = 1 - P_{(4,5),(4,6)} - P_{(4,5),(5,5)} - P_{(4,5),(5,6)} = 0.4037 \tag{44d}$$

These probabilities account for all one-step transitions and sum to one. This result as well as the remaining transition probabilities for other states are calculated in the same manner. These probabilities are used to determine each of the transitions in

the DTMC, resulting in the transition matrix shown in Section 5.3 and Appendix D. A visual representation of the DTMC is shown in Appendix E.

The transition matrix is iterated over 40 time steps with the initial state, $\boldsymbol{\pi_0}$, established at $(1,1)$ so that the probability of a win for each player can be determined. The selection of 40 time steps is made to ensure ample time steps to show convergence to the stationary probabilities while the initial state provides a realistic evaluation of an attacker-defender scenario. As shown in Figure 2, the limit converged at approximately $t = 35$ with $|\mathbf{P}_{i,j}^{35} - \mathbf{P}_{i,j}^{40}| < 0.001$.

For the example given in Appendix D the results of $\boldsymbol{\pi}$ between $t = 35$ and $t = 40$, the following stationary probabilities are found for the Blue win states $(5, s_2)$ for $s_2 \in S_2 \backslash \{6\}$, Red win states $(s_1, 6)$ for $s_1 \in S_1 \backslash \{5\}$, and the stalemate state $(5, 6)$, as shown in Figure 45. Lines that do not converge to zero indicate the stationary probabilities for the absorbing win states.

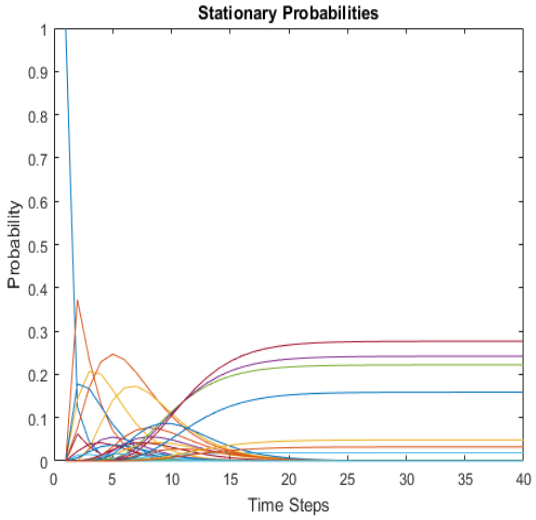



**Figure 2. Stationary Probabilities Over 40 Time Steps**

$$\lim_{t\to\infty} \boldsymbol{\pi_0}\mathbf{P}^t_{i,j} = \boldsymbol{\pi} = \begin{cases} \pi_{(1,6)} = 0.0189 \\ \pi_{(2,6)} = 0.2219 \\ \pi_{(3,6)} = 0.2419 \\ \pi_{(4,6)} = 0.0486 \\ \pi_{(5,1)} = 0.0003 \\ \pi_{(5,2)} = 0.0006 \\ \pi_{(5,3)} = 0 \\ \pi_{(5,4)} = 0.2766 \\ \pi_{(5,5)} = 0.1587 \\ \pi_{(5,6)} = 0.0325 \\ 0, \qquad \text{else} \end{cases} \tag{45}$$

From Equation 45, the first four probabilities listed represent the probability of a Red win, the next five probabilities represent the probability of a Blue win, and the final probability represents the probability of a stalemate. Summing across the stationary probabilities representing each player's win state shows that the game will result in a win for Blue with probability 0.4362, Red with probability 0.5313, and stalemate with probability 0.0325.

Additionally, by setting the initial state vector to be each possible state of the game, excluding the terminating states, the conditional probability of winning for each player is found. Table 18 provides the probability pairing of Blue winning and Red winning, respectively, given each player's current state in the game. For example, if the Blue player is currently in state 1 and the Red player is currently in state 2, the probability of Blue winning is 0.3707 while the probability of Red winning is 0.5993.

These conditional probabilities provide a clear operational view at each moment of the game.

**Table 18. Conditional Player Win Probabilities**

| | | Red State | | | | |
|---|---|---|---|---|---|---|
| | | **1** | **2** | **3** | **4** | **5** |
| **Blue State** | **1** | (0.4362, 0.5313) | (0.3707, 0.5993) | (0.3215, 0.651) | (0.2796, 0.6966) | (0, 1) |
| | **2** | (0.6793, 0.2803) | (0.5077, 0.4563) | (0.3618, 0.6073) | (0.3618, 0.6073) | (0, 1) |
| | **3** | (0.8845, 0.0716) | (0.8664, 0.0863) | (0.8491, 0.1012) | (0.7494, 0.1867) | (0, 1) |
| | **4** | (0.8984, 0.061) | (0.8902, 0.0659) | (0.8902, 0.0659) | (0.8902, 0.0659) | (0.4512, 0.3293) |

Using the techniques presented in Section 4.4, transient analysis is performed. The resulting transient state matrix $\mathbf{Q}$, matrix $\mathbf{M}$ consisting of the expected time periods is state $j$ given the chain starts in state $i$ for all $i, j \in S$, and the matrix $\mathbf{F}$ consisting of probabilities of entering state $j$ given the chain starts in state $i$ for all $i, j \in S$ are presented in Appendices F, G, and H, respectively.

In determining the most likely path of the stochastic game, the matrix $\mathbf{F}$ in conjunction with $\mathbf{P}$ is used by following a path from a given starting state. The path is determined by following the path through states in which the probability of ever entering the state $j$ given the chain started in state $i$ is greatest. Of particular concern is the chain that begins in state $(1, 1)$ as this is the logical start of a cyber attacker-defender scenario given the fact that a defender (Blue) cannot reasonably reach a later state without first detecting the actions of the attacker (Red).

Given that the chain begins at state $(1, 1)$, a transition can be made to state $(1, 2)$, $(2, 1)$, or $(2, 2)$ as observed from $\mathbf{P}$. Of these potential entering states, $\mathbf{F}$ indicates that it is most likely that state $(1, 2)$ will be entered next from $(1, 1)$ with conditional probability 0.6914. Next, state $(1, 2)$ can transition to states $(1, 3)$, $(2, 2)$, or $(2, 3)$. Again referring to $\mathbf{F}$, state $(2, 3)$ is the most likely to be entered with conditional

probability 0.5574. Continuing in this manner indicates that the most likely path for the stochastic game is $(1,1)$, $(1,2)$, $(2,3)$, $(2,4)$, $(3,4)$, then $(4,4)$. Since **F** only considers transient states, it is at this point where the utility of the matrix ends. To determine the final state of the game, **P** is reviewed to identify the most likely end state. From this, it is determined that the game is most likely to conclude in state $(5,4)$ and result in a win for Blue since $P_{(4,4),(5,4)} = 0.375$. Note that among the possible win states from state $(4,4)$ (that is, states $(5,4)$, $(5,5)$, $(5,6)$, and $(4,6)$), state $(5,4)$ has the greatest stationary distribution.

Table 19 provides a summary of the most likely path moving from state $i$ to candidate states state $j_1$, $j_2$, or $j_3$ with the respective conditional probabilities. Entries marked with a * denote the chosen conditional path.

**Table 19. Summary of Most Likely Path**

| **State** $i$ | **State** $j_1 : f_{i,j_1}$ | **State** $j_2 : f_{i,j_2}$ | **State** $j_3 : f_{i,j_3}$ |
|---|---|---|---|
| $(1,1)$ | $*(1,2) : 0.6914$ | $(2,1) : 0.1358$ | $(2,2) : 0.4048$ |
| $(1,2)$ | $(1,3) : 0.4255$ | $(2,2) : 0.4048$ | $*(2,3) : 0.5574$ |
| $(2,3)$ | $*(2,4) : 0.7705$ | $(3,3) : 0.1152$ | $(3,4) : 0.4056$ |
| $(2,4)$ | $(2,5) : 0.2675$ | $*(3,4) : 0.4056$ | $(3,5) : 0.2173$ |
| $(3,4)$ | $(3,5) : 0.2173$ | $*(4,4) : 0.461$ | $(4,5) : 0.1475$ |
| $(4,4)$ | $(4,5) : 0.1475$ | $*(5,4) : 0.375$ | $(5,5) : 0.125$ |

Finding the most likely paths from initial states to "win" states in this manner is useful in prioritizing updates to systems or software. By identifying the most probable path, cyber security analysts can determine where a weak component may lie and assess the feasibility of improving the components performance so that a more desirable path is achieved. In the example above, an analyst may consider making

improvements in sensor performance so that detection may occur sooner with greater chance of success, or improve how resource allocation is assessed to allow for greater accuracy in identifying abnormal behavior.

### Evaluating Sub-Optimal Strategies

Analysis is performed to determine the affect on the results of the operational problem should a player decide to play a sub-optimal strategy in at least one of the tactical games. This is accomplished by changing the strategy played by Blue in $G(1,1)$ to $\alpha^B_{(1,2)}$ and $G(2,1)$ to $\alpha^B_{(2,3)}$, both of which are deviations from the strict PSNE found in each state game and are early in the state sequence progression. A comparison of the stationary distributions found for the initial operational problem and the stationary distributions of the operational problem with sub-optimal strategies is conducted using 40 time steps as the projected point of convergence. The convergence to the stationary probabilities with the sub-optimal Blue strategies is shown in Figure 3 with $|\mathbf{P}^{35}_{i,j} - \mathbf{P}^{40}_{i,j}| < 0.001$. A comparison of the initial stationary probabilities to the sub-optimal Blue strategies probabilities is shown in Table 20.

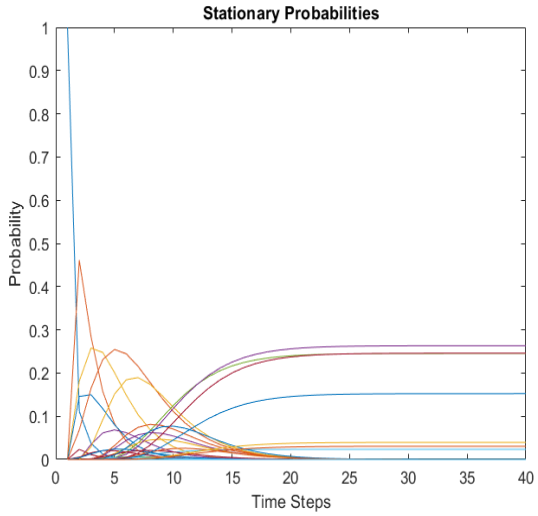



**Figure 3. Stationary Probabilities with Sub-optimal Blue Strategies**

**Table 20. Initial Stationary Probabilities vs Sub-optimal**

| | (1,6) | (2,6) | (3,6) | (4,6) | (5,1) | (5,2) | (5,3) | (5,4) | (5,5) | (5,6) |
|---|---|---|---|---|---|---|---|---|---|---|
| **Initial** | 0.0189 | 0.2219 | 0.2419 | 0.0486 | 0.0003 | 0.0006 | 0 | 0.2766 | 0.1587 | 0.0325 |
| **Sub-Optimal** | 0.0233 | 0.2452 | 0.2632 | 0.0395 | 0 | 0 | 0 | 0.2459 | 0.1523 | 0.0305 |

These sub-optimal strategies result in a Blue win with probability 0.3982, a Red win with probability 0.5713, and stalemate with probability 0.0305. Note that the probability of a Blue win decreased by 0.038 while the probability of a Red win increased by 0.04. While this example provides a relatively small decrease in the probability of a Blue win, the decrease in probability is compounded as Blue continues to play additional sub-optimal strategies while the probability of a Red win will continue to increase. Should Red decide to play sub-optimal strategies, the same effect is observed where Red's probability of winning decreases while Blue's win probability would increase. Thus, the results of playing a sub-optimal strategy show the robustness of playing a Nash equilibrium when the opponent does not.

### Player Policies

Policy analysis is performed from the perspective of the Blue player since it is desired to determine the resiliency of the defender's network. The same policy analysis can be performed from the perspective of the Red player should an analyst wish to evaluate a potential attack strategy.

As previously stated, the original DTMC model utilizes the MSNE from each tactical game to determine a generalized outcome of the operational problem. However, it is unlikely that a player will desire to play strategies based on randomization. In the moment in which the game is played, a decision must be made as to which strategy will be played. Taking the Blue's perspective, it is desired that the probability of a Red win be minimized while maximizing Blue's probability of winning. As such, it

was necessary to minimize Red's expected utility in each stage game whenever possible assuming that the Red player played the mixed strategy as defined by the MSNE. Note that this does not mean that the expected utility of the Blue player increases since the mixed strategy of the Red player remains unchanged. There is usefulness in information operations that make it unclear to the Red player which policy is best by denying information or allowing unclear or ambiguous information about what policy is best.

To accomplish this goal, the following policy is adopted:

- Evaluate each strategy's affect on the opponent's minmax value given the opponent's mixed-strategy. Play the strategy that minimizes the opponent's minmax value.
- If the opponent's minmax value is the same for each strategy, play the strategy with the greatest mixed-strategy probability.
- If the player's mixed strategy is uniformly distributed and no difference among affects on the opponent's minmax value exist, play the strategy that is most preferable, as determined by the player based on factors such as ease of execution and required resources.

To demonstrate, the policy is applied to state game $G(4,5)$ as shown in Table 16. Recall that the mixed strategy of Red is $(0, 0.5, 0.5)$ and the MSNE exists for Blue strategies $\alpha^B_{(4,3)}$ and $\alpha^B_{(4,4)}$. Let $\mu_R(\alpha^B_{(s_1,i)})$ be the minmax value for Red given Blue chooses to play $\alpha^B_{(s_1,i)}$, where $s_1 \in S_1$ and $i = 1, \ldots, N^B_{s_1}$. First setting $P(\alpha^B_{(4,3)}) = 1$ and $P(\alpha^B_{(4,4)}) = 0$,

$$\begin{aligned}\mu_R(\alpha^B_{(4,3)}) &= 0.5u_R(\alpha^B_{(4,3)}, \alpha^R_{(5,2)}) + 0.5u_R(\alpha^B_{(4,3)}, \alpha^R_{(5,3)}) \\ &= 0.5(0.15) + 0.5(0.54) \\ &= 0.345 \end{aligned} \tag{46}$$

Now setting $P(\alpha^B_{(4,3)}) = 0$ and $P(\alpha^B_{(4,4)}) = 1$,

$$\begin{aligned}\mu_R(\alpha^B_{(4,4)}) &= 0.5u_R(\alpha^B_{(4,4)}, \alpha^R_{(5,2)}) + 0.5u_R(\alpha^B_{(4,4)}, \alpha^R_{(5,3)}) \\ &= 0.5(0.4) + 0.5(0.24) \\ &= 0.32 \end{aligned} \tag{47}$$

Since $\mu_R(\alpha^B_{(4,4)}) < \mu_R(\alpha^B_{(4,3)})$, the policy states that Blue should play $\alpha^B_{(4,4)}$.

Continuing in this manner across all state games, the DTMC is reevaluated to observe the difference in the stationary probabilities, thereby determining whether the probability of a Blue win increases. Figure 4 shows the convergence to the stationary probabilities with $|\mathbf{P}^{35}_{i,j} - \mathbf{P}^{40}_{i,j}| < 0.001$, and Table 21 provides a comparison of the initial models stationary probabilities to the Blue policy.

**Table 21. Initial Stationary Probabilities vs Blue Policy**

| | **(1,6)** | **(2,6)** | **(3,6)** | **(4,6)** | **(5,1)** | **(5,2)** | **(5,3)** | **(5,4)** | **(5,5)** | **(5,6)** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Initial** | 0.0189 | 0.2219 | 0.2419 | 0.0486 | 0.0003 | 0.0006 | 0 | 0.2766 | 0.1587 | 0.0325 |
| **Policy** | 0.0189 | 0.2219 | 0.2418 | 0.0478 | 0.0003 | 0.0006 | 0 | 0.2768 | 0.16 | 0.0319 |

Summing the stationary probabilities of each of the Red and Blue win states, it is found that the probability of Red winning is 0.5304, the probability of Blue winning is 0.4377, and the probability of a stalemate is 0.0319. Note that comparing to the

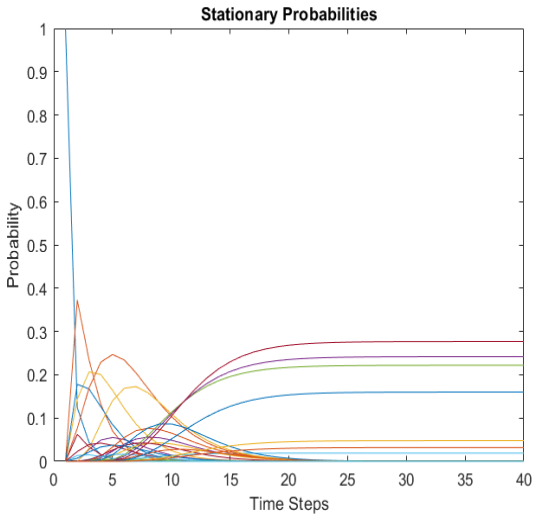



**Figure 4. Stationary Probabilities with Blue Policy**

initial model, the probability of Blue winning increased by 0.0015 and the probability of Red winning decreased by 0.0009. While the difference for each is small, it is determined that the policy is effective in improving Blue's probability of winning for this example and may be more profound given other instances.

## 5.4 Cost Analysis

The established DTMC model is used in conjunction with a vector of baseline damage cost values due to Red actions to determine the expected cost of damage to the Blue player's network in each state. Using the baseline damage costs shown in Table 13 and Equation 36 in Section 4.5, the expected damage costs are found for each state. The matrix of damage costs in thousands of dollars is provided in Appendix I.

Recall from the matrices $\mathbf{F}$ and $\mathbf{P}$ it was determined that the most likely path of the operational problem given the chain begins in state $(1,1)$ was $(1,1)$, $(1,2)$, $(2,3)$, $(2,4)$, $(3,4)$, $(4,4)$, and $(5,4)$. The associated damage cost for each state is shown

in Table 22. Summing across the row of damage costs indicates that the expected damage cost for the path is $21,536.

**Table 22. Damage Cost of Most Likely Path**

| **State** | $(1,1)$ | $(1,2)$ | $(2,3)$ | $(2,4)$ | $(3,4)$ | $(4,4)$ |
|---|---|---|---|---|---|---|
| **Damage Cost** | 1.2346 | 4.2545 | 11.1477 | 2.3912 | 0.7375 | 1.7701 |

To demonstrate how both the most likely path and the expected damage cost can be affected by changes in the Blue player's utilities, a scenario is explored in which the Blue player must decide whether to repair or upgrade existing components that affect the utilities of each of the Blue player's actions given a specified monetary threshold for resource improvement allocation. It is assumed that should a component be repaired, it is returned to full capacity and the initial utility is not affected. However, if there are insufficient funds for repairing or upgrading a component, no action is taken causing a degradation in the effectiveness for the given component. An integer program model is developed that considers the MSNE of each state game, the baseline damage cost from each of Red's strategies, the Blue strategic form game matrix for the operational problem consisting of the initial utilities, the Blue strategic form game matrix for the operational problem consisting of utilities if components upgraded, and the cost of repairing and upgrading each component. The integer program used is provided in Section 4.5.

The optimal solution of the integer program indicates that most components should receive an upgrade with few exceptions. A summary of the optimal solution and the decision made for each action is shown in Table 23. Note that all cost values are in thousands of dollars.

It is clear from the table that the damage cost, consisting of the product of the operational problem MSNE, expected damage cost from actions taken by the Red player, and the probability of failure of the Blue player's action, has a great affect on the expected cost associated with each action.

A sensitivity analysis between the cost of the decision made for each of the Blue player's actions and the alternative decisions is provided in Table 24. The difference of these values provide bounds on the expected cost in which a change in decision may be potentially be made. Note that negative difference values indicate the amount in which the allocation threshold would need to decrease before a potential no action decision for the given component.

**Table 23. IP Optimal Solution Summary**

| Action | Decision | Baseline Cost | Damage Cost | Expected Cost |
|---|---|---|---|---|
| $\alpha^B_{(1,1)}$ | No Action | 0 | 6.27144 | 6.27144 |
| $\alpha^B_{(1,2)}$ | Upgrade | 2 | 6.9772 | 8.9772 |
| $\alpha^B_{(2,1)}$ | Upgrade | 1 | 2.2 | 3.2 |
| $\alpha^B_{(2,2)}$ | Upgrade | 2 | 7.2 | 9.2 |
| $\alpha^B_{(2,3)}$ | Upgrade | 4 | 4.125 | 8.125 |
| $\alpha^B_{(3,1)}$ | No Action | 0 | 0.3 | 0.3 |
| $\alpha^B_{(3,2)}$ | No Action | 0 | 4.56703 | 4.56703 |
| $\alpha^B_{(3,3)}$ | Repair | 0.75 | 3.57932 | 4.32932 |
| $\alpha^B_{(3,4)}$ | Repair | 1.5 | 0.4573 | 1.9573 |
| $\alpha^B_{(3,5)}$ | Repair | 2 | 5.25006 | 7.25006 |
| $\alpha^B_{(4,1)}$ | Upgrade | 1.3 | 3 | 4.3 |
| $\alpha^B_{(4,2)}$ | Upgrade | 1 | 3.5 | 4.5 |
| $\alpha^B_{(4,3)}$ | Upgrade | 1 | 2.19575 | 3.19575 |
| $\alpha^B_{(4,4)}$ | Upgrade | 1 | 2.58575 | 3.58575 |

Given the observation of the affect of the damage cost on the total expected cost, it is likely that most of the decisions recommended for this instance of the IP could shift if the alternative solution were to provide a lesser probability of failure against each of the Red player's actions without increasing the baseline cost. The utilities

**Table 24. IP Sensitivity Analysis**

| **Action** | **Alternative 1 Cost** | **Alternative 2 Cost** | **Decision Cost** | **Difference From 1** | **Diffference From 2** |
|---|---|---|---|---|---|
| $\alpha^B_{(1,1)}$ | 9.7228 | 12.2296 | 6.27144 | 3.45136 | 5.95816 |
| $\alpha^B_{(1,2)}$ | 7.7772 | 6.02856 | 8.9772 | -1.2 | -2.94864 |
| $\alpha^B_{(2,1)}$ | 2.35 | 1.38 | 3.2 | -0.85 | -1.82 |
| $\alpha^B_{(2,2)}$ | 10.0375 | 7.135 | 9.2 | 0.8375 | -2.065 |
| $\alpha^B_{(2,3)}$ | 9.2875 | 5.625 | 8.125 | 1.1625 | -2.5 |
| $\alpha^B_{(3,1)}$ | 2.8 | 6.3 | 0.3 | 2.5 | 6 |
| $\alpha^B_{(3,2)}$ | 9.22964 | 11.7926 | 4.56703 | 4.66261 | 7.22557 |
| $\alpha^B_{(3,3)}$ | 5.58976 | 3.59394 | 4.32932 | 1.26044 | -0.73538 |
| $\alpha^B_{(3,4)}$ | 8.46437 | 0.4737 | 1.9573 | 6.50707 | -1.4836 |
| $\alpha^B_{(3,5)}$ | 9.64293 | 5.83219 | 7.25006 | 2.39287 | -1.41787 |
| $\alpha^B_{(4,1)}$ | 4.25 | 2.96 | 4.3 | -0.05 | -1.34 |
| $\alpha^B_{(4,2)}$ | 5 | 3.8 | 4.5 | 0.5 | -0.7 |
| $\alpha^B_{(4,3)}$ | 3.261 | 2.2958 | 3.19575 | 0.06525 | -0.89995 |
| $\alpha^B_{(4,4)}$ | 3.639 | 2.7242 | 3.58575 | 0.05325 | -0.86155 |

for the actions that correspond with each of the upgraded components are used to provide an updated DTMC model to determine if a better conditional path can be found while decreasing the expected damage cost associated with the path from the initial solution to the operational network security problem, as well as assessing the inherent risk given the decision made for each component.

For the updated model, each state game is solved with the updated utilities in the applicable state games, the results of which are found in Appendix C. Using the same methods as before, the DTMC is evaluated with the new Nash equilibria. A comparison of the probabilities to the initial model and the resulting convergence of the stationary probabilities with $|\mathbf{P}_{i,j}^{35} - \mathbf{P}_{i,j}^{40}| < 0.001$ are shown in Figure 5 and Table 25, respectively.

**Table 25. Initial Stationary Probabilities vs Allocation Utilities**

| | **(1,6)** | **(2,6)** | **(3,6)** | **(4,6)** | **(5,1)** | **(5,2)** | **(5,3)** | **(5,4)** | **(5,5)** | **(5,6)** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Initial** | 0.0189 | 0.2219 | 0.2419 | 0.0486 | 0.0003 | 0.0006 | 0 | 0.2766 | 0.1587 | 0.0325 |
| **Allocation** | 0.0362 | 0.1268 | 0.2567 | 0.035 | 0.0014 | 0.0017 | 0 | 0.3068 | 0.1991 | 0.0362 |

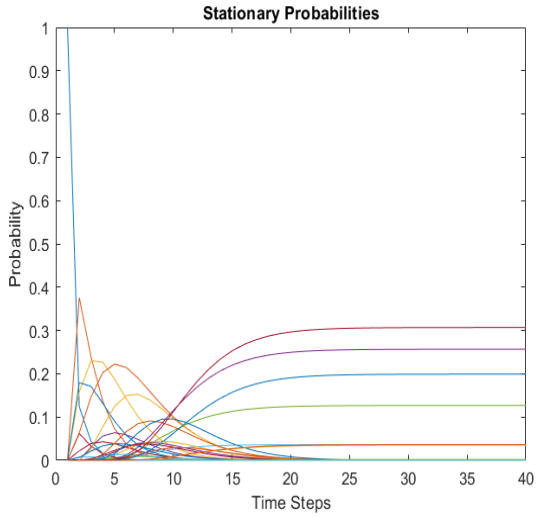



**Figure 5. Stationary Probabilities with Allocation Utilities**

From Table 25, the first four columns show the comparison of the probabilities for a Red win, the next five columns show the comparison of the probabilities for a Blue win, and the final column shows the comparison of the probability for a stalemate. Thus, summing across the respective columns for the row of stationary probabilities

from the updated tactical game utilities, the probability of a Blue win is 0.5452, the probability of a Red win is 0.4547, and the probability of stalemate is 0.0362. Note that Blue's probability of winning increases by 0.109 and Red's probability of winning decreases by 0.0766. These changes are not relatively large, however Blue's win probability will continue to increase should additional funds become available for resource improvement.

Evaluating the updated matrices $\mathbf{F}$ and $\mathbf{P}$, it is found that the most likely path of the game given the chain starts in state $(1,1)$ is $(1,1)$, $(1,2)$, $(2,3)$, $(3,3)$, $(4,4)$, and $(5,4)$. A summary of the likely path and conditional probabilities is shown in Table 26. Each candidate transition state selected for the likely path is indicated by an *.

**Table 26. Summary of Allocation Most Likely Path**

| **State** $i$ | **State** $j_1 : f_{i,j_1}$ | **State** $j_2 : f_{i,j_2}$ | **State** $j_3 : f_{i,j_3}$ |
|---|---|---|---|
| $(1,1)$ | $*(1,2) : 0.6914$ | $(2,1) : 0.1358$ | $(2,2) : 0.3918$ |
| $(1,2)$ | $(1,3) : 0.4756$ | $(2,2) : 0.3918$ | $*(2,3) : 0.5153$ |
| $(2,3)$ | $*(2,4) : 0.7341$ | $(3,3) : 0.1199$ | $(3,4) : 0.4825$ |
| $(2,4)$ | $(2,5) : 0.1928$ | $*(3,4) : 0.4825$ | $(3,5) : 0.225$ |
| $(3,4)$ | $(3,5) : 0.225$ | $*(4,4) : 0.5114$ | $(4,5) : 0.1681$ |
| $(4,4)$ | $(4,5) : 1339$ | $*(5,4) : 0.37$ | $(5,5) : 1384$ |

The expected damage cost per state within the most likely path is shown in Table 27 with all values in thousands of dollars. Summing each of the damage cost values gives a total expected damage cost of $20,834 for the likely path with the updated utilities, resulting in a decrease of $702 from the initial solution of the operational problem.

**Table 27. Damage Cost of Allocation Most Likely Path**

| **State** | $(1,1)$ | $(1,2)$ | $(2,3)$ | $(2,4)$ | $(3,4)$ | $(4,4)$ |
|---|---|---|---|---|---|---|
| **Damage Cost** | 1.2963 | 4.53 | 10.3061 | 1.7619 | 0.8774 | 2.062 |

Thus, the decision made for this instance based on the optimal solution of the IP provides an improvement on the initial operational network security problem for the probability of winning for the Blue player, the conditional path, and the associated total expected cost of damages. This may not always be the case and may change substantially if the cost threshold for resource allocation is adjusted.

# VI. Conclusions and Future Research

## 6.1 Conclusions

This research provides a novel approach to cyber system analysis and network improvement resource allocation by developing a stochastic game theoretical model for evaluating the resiliency of a defender's network. The operational network security problem is solved by implementing game theory to determine the solution of 20 tactical normal form game sub-problems, each of which consist of subsets of 14 defender (Blue) player strategies and 21 attacker (Red) player strategies. The Nash equilibria from each of these state games are used to determine the state transitional probabilities in a discrete-time Markov chain, in turn providing the probability of a Blue win and Red win determined by the stationary distribution of the Markov chain. In turn, transient analysis provides an evaluation of the expected damage cost due to each Red attack strategy. Information gained from the normal form games and the Markov chain are utilized in an integer program to advise the Blue player on repairing or upgrading networks components that are used in each of the Blue strategies. The results from the integer program provide an updated Markov chain that indicate an improvement in the probability of a Blue win given that Blue performs the recommended decisions.

The results of the research provide answers to the following questions, as originally posed in Chapter I.

**Question 1: How can the resiliency of a cyber system be evaluated using game theoretical and stochastic modeling?**

The possible actions to be taken by the Blue and Red actors are specified. These lists are constructed in such a way that they are only as specific as needed so that

analysis can be performed. For example, instead of specifying the inspection of intrusion detection or intrusion prevention systems, a generalized action of inspection of sensor devices, as done in this research, can be used that encompasses both of these devices as well as any other components used in this action.

Next, the actions should be grouped together as states such that the states follow in a logical order through the desired attack and defend posturing. This enables a progression through an attacker-defender scenario from detection and scanning for the defender and attacker, respectively, to the conclusion of the scenario in which either the defender or attacker may "win." Each state can then be formed into a normal form game based on the actions taken in each state so that the success of each player can be determined in the form of Nash equilibria.

Finally, the construction and evaluation of the state games provides a natural segue to Markov chains in which the mixed strategies of each player and state game can be used as weights on the utilities corresponding with the Nash equilibria, thereby providing the probability of success in any given state for each player. These success probabilities form the basis of the state transition probabilities, enabling the construction of a transition matrix in which stationary probabilities indicate the probability of each player winning the stochastic game over a discrete time interval.

**Question 2: Given the current state of a cyber system and potential actions of a malicious actor, how may cyber security analysts evaluate the likelihood of success or failure of defensive cyber actions?**

Stationary probabilities of a discrete time Markov chain formed from the state games provide a determination of the probability of winning for each player. The success and failure of a potential strategy policy over the course of the stochastic

game is found via the convergence of the transition matrix to each player's win states over a discrete set of time steps.

**Question 3: In what manner can a cost analysis be performed on an upgrade or replacement of cyber defense hardware and software while in turn evaluating the effects on the system?**

The results of the state games can be utilized as data points within an integer program to provide a cost analysis of upgrading or replacing cyber defense hardware and software. The integer program should consist of each of the total costs for upgrading and replacing components, the utilities of the Blue player for each of the decisions, the Nash equilibria found from state game analysis, and an estimated damage cost due to the actions performed by the Red player against the Blue network. The results of the mathematical programming model are used to update the properties of the discrete-time Markov chain, providing the degree of improvement in the resiliency of the Blue player's network given the resource allocation decision taken. Furthermore, the developed formulation is highly scalable and allows analysts to adapt the model according to their network needs.

## 6.2 Future Research

While this research proposes a model in which each of the stated research objectives are met, greater fidelity can be gained by considering alterations to the model. This section proposes such alterations should a researcher wish to continue the development of the proposed stochastic game model.

This research uses a single instance of the operational problem with selected alterations to illustrate concepts like sub-optimal player strategies and stability of Nash equilibria. The significance of the model can be explored further by experimenting

on alternative instances of the game. These experiments may consider alternative pricing on the decision to upgrade or replace components or changes in each player's utilities in the tactical games.

Attack graphs provide analysis on the likelihood of a given attack strategy taken by a malicious actor. The model formed in this research lends well to the use of attack graphs, however the specificity of the attacker's actions and the inability to bypass or step backwards in states make the direct translation to an attack graph infeasible.

To correct this, one may consider the specific actions that may be taken for each of the attacker's strategies and determine which of these actions allow for a win for the attacker without the necessity of entering each state.

Another point of improvement on this model would be to consider an additional strategy to take no action. This strategy should be made available to either player at any point in the game. Doing so allows the player to observe the opponent, gain knowledge of the opponent's strategies, and update the knowledge of the game so that a better informed strategy decision can be made. When considering this strategy for the cost evaluation, a discount factor to the Blue utilities should be applied to account for degradation of components due to inaction.

Additionally, this model does not allow either player to transition to an earlier state in the Markov chain. This was a necessary assumption so that a foundational model could be constructed, however this is not how a real-world attacker-defender scenario would proceed. At any point in either player's sequence, they may be able to perform an action that negates the progress made by the opponent. A model that accounts for this possibility would provide better insight on the interaction between players over the course of the game.

Finally, it is unlikely that either player would have complete knowledge of their opponents utility for each strategy. As such, a factor of uncertainty in the form

of bounded utilities would be appropriate. In doing so, one will need to consider the instability of Nash equilibria and equilibria could be represented to inform the Markov chain.

# Appendix A. Tactical Normal Form State Games

| Blue/Red | Pinging | Channel Monitoring | Traffic Monitoring | Open Source | Trojan Horse | Spoofing | Obtain Credentials | Inject | Overflow |
|---|---|---|---|---|---|---|---|---|---|
| Sensor Data | 0.2, 0.56 | 0.1, 0.27 | 0.2, 0.4 | 0.1, 0.36 | 0.2, 0.4 | 0.3, 0.21 | 0, 0.4 | 0, 0.2 | 0.3, 0.35 |
| Sniffers | 0.1, 0.63 | 0.2, 0.24 | 0.3, 0.35 | 0.05, 0.38 | 0.2, 0.4 | 0.3, 0.21 | 0, 0.4 | 0.2, 0.16 | 0.3, 0.35 |
| Check Permissions | 0, 0.7 | 0, 0.3 | 0, 0.5 | 0.1, 0.36 | 0, 0.5 | 0.1, 0.27 | 0.4, 0.24 | 0.4, 0.12 | 0, 0.5 |
| Resource Allocation | 0.3, 0.49 | 0.1, 0.27 | 0.2, 0.4 | 0, 0.4 | 0, 0.5 | 0.2, 0.24 | 0.2, 0.32 | 0.1, 0.18 | 0.3, 0.35 |
| Signatures | 0, 0.7 | 0, 0.3 | 0, 0.5 | 0, 0.4 | 0.2, 0.4 | 0, 0.3 | 0, 0.4 | 0.2, 0.16 | 0.1, 0.45 |
| Firewall Rules | 0.3, 0.49 | 0.4, 0.18 | 0.5, 0.25 | 0.2, 0.32 | 0.2, 0.4 | 0.4, 0.18 | 0, 0.4 | 0.3, 0.14 | 0.7, 0.15 |
| Network Reconfiguration | 0.5, 0.35 | 0.7, 0.09 | 0.7, 0.15 | 0.1, 0.36 | 0, 0.5 | 0.6, 0.12 | 0.1, 0.36 | 0.1, 0.18 | 0.6, 0.2 |
| Password Policies | 0, 0.7 | 0, 0.3 | 0, 0.5 | 0.6, 0.16 | 0, 0.5 | 0, 0.3 | 0.7, 0.12 | 0, 0.2 | 0, 0.5 |
| Port/Service Management | 0.4, 0.42 | 0.3, 0.21 | 0.7, 0.28 | 0.3, 0.28 | 0.1, 0.45 | 0.5, 0.15 | 0, 0.4 | 0.1, 0.18 | 0.2, 0.4 |
| Patching | 0, 0.7 | 0, 0.3 | 0, 0.5 | 0, 0.4 | 0.3, 0.35 | 0, 0.3 | 0, 0.4 | 0, 0.2 | 0.7, 0.15 |
| Logs | 0.05, 0.665 | 0, 0.3 | 0, 0.5 | 0, 0.4 | 0, 0.5 | 0.1, 0.27 | 0.5, 0.2 | 0.2, 0.16 | 0, 0.5 |
| Hashes | 0, 0.7 | 0, 0.3 | 0, 0.5 | 0, 0.4 | 0.4, 0.3 | 0, 0.3 | 0.4, 0.24 | 0, 0.2 | 0, 0.5 |
| Directories | 0, 0.7 | 0, 0.3 | 0, 0.5 | 0, 0.4 | 0, 0.5 | 0, 0.3 | 0, 0.4 | 0, 0.2 | 0, 0.5 |
| Files | 0, 0.7 | 0, 0.3 | 0, 0.5 | 0, 0.4 | 0, 0.5 | 0, 0.3 | 0, 0.4 | 0, 0.2 | 0, 0.5 |

| Blue/Red | Hijacking | Packet Manipulation | Flood | Process Manipulation | Malware | Covert Channels | Rootkit | Spyware | Backdoor |
|---|---|---|---|---|---|---|---|---|---|
| Sensor Data | 0.2, 0.24 | 0.5, 0.1 | 0.7, 0.09 | 0.4, 0.12 | 0.6, 0.16 | 0.6, 0.04 | 0.6, 0.12 | 0.7, 0.15 | 0.5, 0.15 |
| Sniffers | 0.2, 0.24 | 0.4, 0.12 | 0, 0.3 | 0.3, 0.14 | 0.4, 0.24 | 0.3, 0.07 | 0.3, 0.21 | 0.4, 0.3 | 0.3, 0.21 |
| Check Permissions | 0, 0.3 | 0, 0.2 | 0, 0.3 | 0, 0.2 | 0, 0.4 | 0, 0.1 | 0.6, 0.12 | 0, 0.5 | 0.1, 0.27 |
| Resource Allocation | 0, 0.3 | 0.2, 0.16 | 0.8, 0.06 | 0.7, 0.06 | 0.2, 0.32 | 0.1, 0.09 | 0, 0.3 | 0, 0.5 | 0.4, 0.18 |
| Signatures | 0, 0.3 | 0.4, 0.12 | 0.5, 0.15 | 0.4,0.12 | 0.7, 0.12 | 0.5, 0.05 | 0.4, 0.18 | 0.4, 0.3 | 0.5, 0.15 |
| Firewall Rules | 0, 0.3 | 0.1, 0.18 | 0.4, 0.18 | 0.1, 0.18 | 0.3, 0.28 | 0.4, 0.06 | 0, 0.3 | 0, 0.5 | 0, 0.3 |
| Network Reconfiguration | 0.4, 0.18 | 0.5, 0.1 | 0.3, 0.21 | 0.3, 0.14 | 0.3, 0.28 | 0.4, 0.06 | 0.2, 0.24 | 0.2, 0.4 | 0, 0.3 |
| Password Policies | 0.1, 0.27 | 0, 0.2 | 0, 0.3 | 0, 0.2 | 0, 0.4 | 0, 0.1 | 0.2, 0.24 | 0, 0.5 | 0, 0.3 |
| Port/Service Management | 0.3, 0.21 | 0, 0.2 | 0.6, 0.12 | 0.4, 0.12 | 0.3, 0.28 | 0.2, 0.08 | 0.6, 0.12 | 0.6, 0.2 | 0.5, 0.15 |
| Patching | 0, 0.3 | 0.8, 0.04 | 0.2, 0.24 | 0.6, 0.08 | 0.8, 0.08 | 0.7, 0.03 | 0.8, 0 | 0.7, 0.15 | 0.5, 0.15 |
| Logs | 0.05, 0.285 | 0.1, 0.18 | 0, 0.3 | 0.2, 0.16 | 0, 0.4 | 0.1, 0.09 | 0.1, 0.27 | 0, 0.5 | 0.2, 0.24 |
| Hashes | 0, 0.3 | 0, 0.2 | 0, 0.3 | 0.3, 0.14 | 0, 0.4 | 0, 0.1 | 0.1, 0.27 | 0.5, 0.25 | 0.5, 0.15 |
| Directories | 0, 0.3 | 0, 0.2 | 0, 0.3 | 0.2, 0.16 | 0, 0.4 | 0, 0.1 | 0, 0.3 | 0, 0.5 | 0, 0.3 |
| Files | 0, 0.3 | 0, 0.2 | 0, 0.3 | 0.5, 0.1 | 0, 0.4 | 0, 0.1 | 0, 0.3 | 0, 0.5 | 0, 0.3 |

| Blue/Red | Alter Logs | Hidden Directories | Hidden Files |
|---|---|---|---|
| Sensor Data | 0, 0.2 | 0, 0.5 | 0, 0.6 |
| Sniffers | 0, 0.2 | 0, 0.5 | 0, 0.6 |
| Check Permissions | 0.1, 0.18 | 0.1, 0.45 | 0.1, 0.54 |
| Resource Allocation | 0, 0.2 | 0, 0.5 | 0, 0.6 |
| Signatures | 0, 0.2 | 0, 0.5 | 0, 0.6 |
| Firewall Rules | 0, 0.2 | 0, 0.5 | 0, 0.6 |
| Network Reconfiguration | 0.2, 0.16 | 0, 0.5 | 0, 0.6 |
| Password Policies | 0, 0.2 | 0, 0.5 | 0, 0.6 |
| Port/Service Management | 0, 0.2 | 0, 0.5 | 0, 0.6 |
| Patching | 0, 0.2 | 0, 0.5 | 0, 0.6 |
| Logs | 0.5, 0.1 | 0.3, 0.35 | 0.3, 0.42 |
| Hashes | 0.1, 0.18 | 0.1, 0.45 | 0.1, 0.54 |
| Directories | 0.3, 0.14 | 0.7, 0.15 | 0.1, 0.54 |
| Files | 0.1, 0.18 | 0.2, 0.4 | 0.6, 0.24 |

# Appendix B. Pure and Mixed-Strategy Nash Equilibria

| State | PSNE | MSNE |
|---|---|---|
| (1,1) | $\alpha^B_{(1,1)}, \alpha^R_{(1,1)}$ | $(1,0),(1,0,0,0)$ |
| (1,2) | $\alpha^B_{(1,1)}, \alpha^R_{(2,1)}$<br>$\alpha^B_{(1,2)}, a^R_{(2,1)}$ | $(0.5,0.5),(1,0,0,0,0)$ |
| (1,3) | $\alpha^B_{(1,1)}, \alpha^R_{(3,1)}$<br>$\alpha^B_{(1,2)}, \alpha^R_{(3,1)}$ | $(0.286,0.714),(1,0,0,0,0)$ |
| (1,4) | $\alpha^B_{(1,1)}, \alpha^R_{(4,3)}$<br>$\alpha^B_{(1,1)}, \alpha^R_{(4,4)}$ | $(1,0),(0,0,0.5,0.5)$ |
| (1,5) | $\alpha^B_{(1,1)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(1,2)}, \alpha^R_{(5,3)}$ | $(0.5,0.5),(0,0,1)$ |
| (2,1) | $\alpha^B_{(2,2)}, \alpha^R_{(1,1)}$ | $(0,1,0),(1,0,0,0)$ |
| (2,2) | Does Not Exist | $(0,0.25,0.75),(0.5,0,0,0,0.5)$ |
| (2,3) | Does Not Exist | $(0,0.9,0.1),(1,0,0,0,0)$<br>$(0.643,0,0.357),(1,0,0,0,0)$<br>$(0,0,1),(1,0,0,0,0)$ |
| (2,4) | $\alpha^B_{(2,3)}, \alpha^R_{(4,3)}$ | $(0,0,1),(0,0,1,0)$ |
| (2,5) | $\alpha^B_{(2,1)}, \alpha^R_{(5,3)}$ | $(1,0,0),(0,0,1)$ |

| State | PSNE | MSNE |
|---|---|---|
| (3,1) | Does Not Exist | $(0, 0.93, 0, 0.07, 0), (0.67, 0, 0, 0.33)$ |
| (3,2) | Does Not Exist | $(0, 0, 0.116, 0, 0.884), (0.7, 0, 0.3, 0, 0)$ |
| (3,3) | Does Not Exist | $(0, 0.6749, 0, 0.0149, 0.3102), (0.455, 0, 0.152, 0, 0.393)$ |
| (3,4) | $\alpha^B_{(3,5)}, \alpha^R_{(4,3)}$<br>$\alpha^B_{(3,5)}, \alpha^R_{(4,4)}$ | $(0, 0, 0, 0, 1), (0, 0, 0.5, 0.5)$ |
| (3,5) | $\alpha^B_{(3,1)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(3,2)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(3,3)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(3,4)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(3,5)}, \alpha^R_{(5,3)}$ | $(0.2, 0.2, 0.2, 0.2, 0.2), (0, 0, 1)$ |
| (4,1) | $\alpha^B_{(4,1)}, \alpha^R_{(1,1)}$ | $(1, 0, 0, 0), (1, 0, 0, 0)$ |
| (4,2) | $\alpha^B_{(4,1)}, \alpha^R_{(2,5)}$<br>$\alpha^B_{(4,2)}, \alpha^R_{(2,5)}$<br>$\alpha^B_{(4,3)}, \alpha^R_{(2,5)}$<br>$\alpha^B_{(4,4)}, \alpha^R_{(2,5)}$ | $(0.25, 0.25, 0.25, 0.25), (0, 0, 0, 0, 1)$ |
| (4,3) | $\alpha^B_{(4,1)}, \alpha^R_{(3,5)}$<br>$\alpha^B_{(4,2)}, \alpha^R_{(3,5)}$<br>$\alpha^B_{(4,3)}, \alpha^R_{(3,5)}$<br>$\alpha^B_{(4,4)}, \alpha^R_{(3,5)}$ | $(0.25, 0.25, 0.25, 0.25), (0, 0, 0, 0, 1)$ |
| (4,4) | Does Not Exist | $(0, 1, 0, 0), (0, 0, 1, 0)$<br>$(0.08, 0.92, 0, 0), (0, 0, 1, 0)$ |
| (4,5) | Does Not Exist | $(0, 0, 0.29, 0.71), (0, 0.5, 0.5)$ |

# Appendix C. Nash Equilibria from Allocation IP

| State | PSNE | MSNE |
|---|---|---|
| (1,1) | $\alpha^B_{(1,2)}, \alpha^R_{(1,1)}$ | $(0,1),(1,0,0,0)$ |
| (1,2) | $\alpha^B_{(1,1)}, \alpha^R_{(2,1)}$ | $(1,0),(1,0,0,0,0)$ |
| (1,3) | Does Not Exist | $(0.33,0.66),(0.9333,0,0.0667,0,0)$ |
| (1,4) | $\alpha^B_{(1,1)}, \alpha^R_{(4,3)}$ | $(1,0),(0,0,1,0)$ |
| (1,5) | $\alpha^B_{(1,1)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(1,2)}, \alpha^R_{(5,3)}$ | $(0.5,0.5),(0,0,1)$ |
| (2,1) | $\alpha^B_{(2,2)}, \alpha^R_{(1,1)}$ | $(0,1,0),(1,0,0,0)$ |
| (2,2) | Does Not Exist | $(0.3125,0,0.6875),(0.4286,0,0.5714,0,0)$<br>$(0,0.3125,0.6875),(0.4286,0,0.5714,0,0)$ |
| (2,3) | $\alpha^B_{(2,2)}, \alpha^R_{(3,1)}$<br>$\alpha^B_{(2,3)}, \alpha^R_{(3,1)}$ | $(0.375,0.625,0),(1,0,0,0,0)$<br>$(0.5833,0,0.4167),(1,0,0,0,0)$<br>$(0,0.5,0.5),(1,0,0,0,0)$ |
| (2,4) | $\alpha^B_{(2,3)}, \alpha^R_{(4,3)}$ | $(0,0,1),(0,0,1,0)$ |
| (2,5) | $\alpha^B_{(2,1)}, \alpha^R_{(5,3)}$ | $(1,0,0),(0,0,1)$ |
| (3,1) | $\alpha^B_{(3,2)}, \alpha^R_{(1,1)}$<br>$\alpha^B_{(3,2)}, \alpha^R_{(1,3)}$<br>$\alpha^B_{(3,4)}, \alpha^R_{(1,1)}$<br>$\alpha^B_{(3,4)}, \alpha^R_{(1,3)}$ | $(0,0.5,0,0.5,0),(0.5,0,0.5,0)$<br>$(0,0.2857,0,0.7143,0),(1,0,0,0)$ |

| **State** | **PSNE** | **MSNE** |
|---|---|---|
| (3,2) | Does Not Exist | $(0, 0, 0.0685, 0, 0.9315), (0.7, 0, 0.3, 0, 0)$ |
| (3,3) | Does Not Exist | $(0, 0, 0, 0.7586, 0.2414), (0.625, 0, 0, 0, 0.375)$ |
| (3,4) | $\alpha^B_{(3,5)}, \alpha^R_{(4,3)}$<br>$\alpha^B_{(3,5)}, \alpha^R_{(4,4)}$ | $(0, 0, 0, 0, 1), (0, 0, 0.5, 0.5)$ |
| (3,5) | $\alpha^B_{(3,1)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(3,2)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(3,3)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(3,4)}, \alpha^R_{(5,3)}$<br>$\alpha^B_{(3,5)}, \alpha^R_{(5,3)}$ | $(0.2, 0.2, 0.2, 0.2, 0.2), (0, 0, 1)$ |
| (4,1) | $\alpha^B_{(4,1)}, \alpha^R_{(1,1)}$ | $(1, 0, 0, 0), (1, 0, 0, 0)$ |
| (4,2) | $\alpha^B_{(4,1)}, \alpha^R_{(2,5)}$<br>$\alpha^B_{(4,2)}, \alpha^R_{(2,5)}$<br>$\alpha^B_{(4,3)}, \alpha^R_{(2,5)}$<br>$\alpha^B_{(4,4)}, \alpha^R_{(2,5)}$ | $(0.25, 0.25, 0.25, 0.25), (0, 0, 0, 0, 1)$ |
| (4,3) | $\alpha^B_{(4,1)}, \alpha^R_{(3,5)}$<br>$\alpha^B_{(4,2)}, \alpha^R_{(3,5)}$<br>$\alpha^B_{(4,3)}, \alpha^R_{(3,5)}$<br>$\alpha^B_{(4,4)}, \alpha^R_{(3,5)}$ | $(0.25, 0.25, 0.25, 0.25), (0, 0, 0, 0, 1)$ |
| (4,4) | $\alpha^B_{(4,2)}, \alpha^R_{(4,3)}$ | $(0, 1, 0, 0), (0, 0, 1, 0)$ |
| (4,5) | Does Not Exist | $(0, 0, 0.5077, 0.4923), (0, 0.4167, 0.5833)$ |

# Appendix D. Transition Matrix Partitions

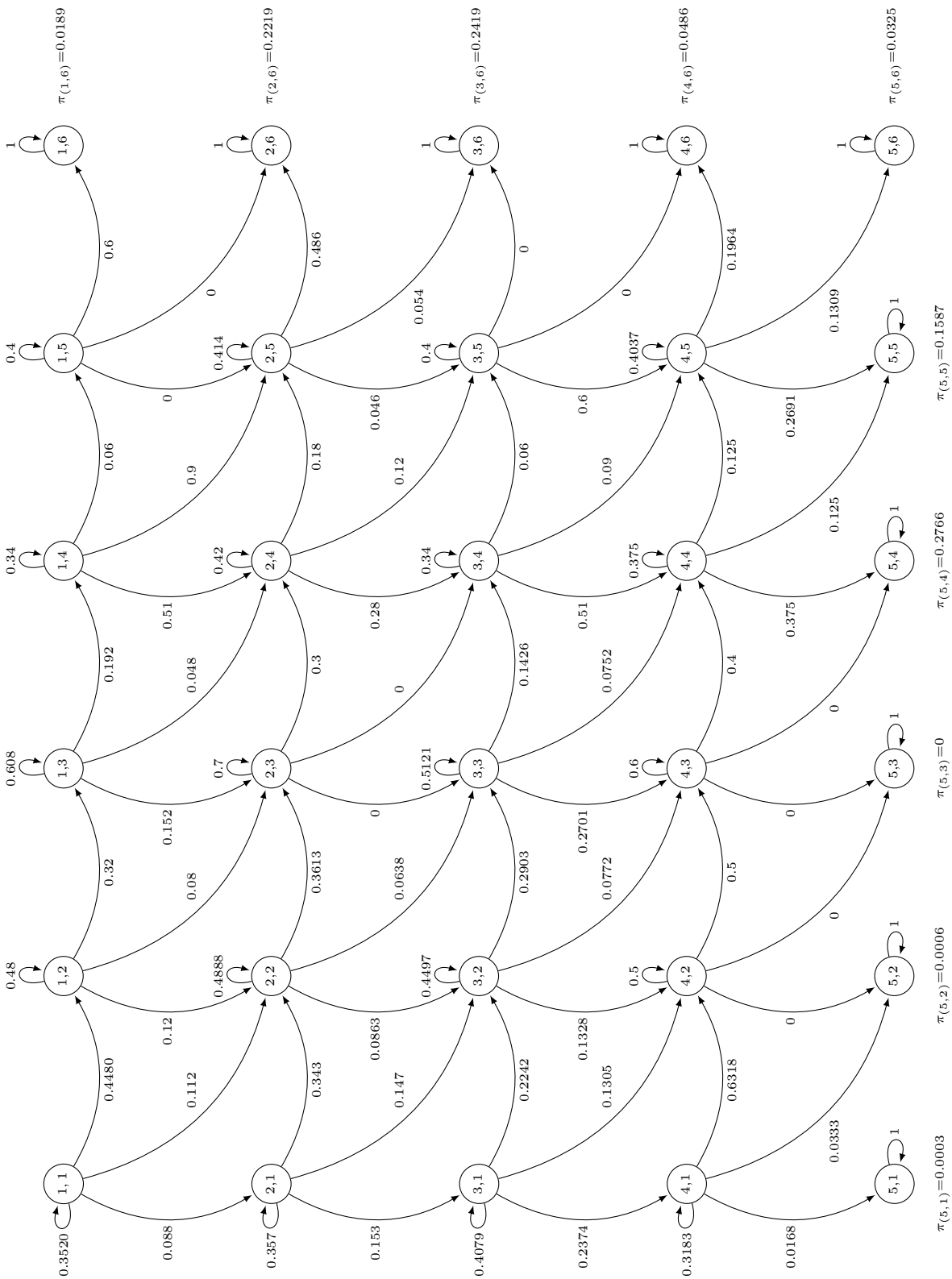
$$\mathbf{P}_1 = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (1,4) \\ (1,5) \\ (1,6) \end{array} \begin{array}{c} \begin{array}{cccccccccccc} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (1,6) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) & (2,6) \end{array} \\ \begin{pmatrix} 0.352 & 0.448 & 0 & 0 & 0 & 0 & 0.088 & 0.1120 & 0 & 0 & 0 & 0 \\ 0 & 0.48 & 0.32 & 0 & 0 & 0 & 0 & 0.12 & 0.08 & 0 & 0 & 0 \\ 0 & 0 & 0.608 & 0.192 & 0 & 0 & 0 & 0 & 0.152 & 0.048 & 0 & 0 \\ 0 & 0 & 0 & 0.34 & 0.06 & 0 & 0 & 0 & 0 & 0.51 & 0.09 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{P}_2 = \begin{array}{c} \\ (2,1) \\ (2,2) \\ (2,3) \\ (2,4) \\ (2,5) \\ (2,6) \end{array} \begin{array}{c} \begin{array}{cccccccccccc} (2,1) & (2,2) & (2,3) & (2,4) & (2,5) & (2,6) & (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (3,6) \end{array} \\ \begin{pmatrix} 0.357 & 0.343 & 0 & 0 & 0 & 0 & 0.153 & 0.147 & 0 & 0 & 0 & 0 \\ 0 & 0.4888 & 0.3613 & 0 & 0 & 0 & 0 & 0.0863 & 0.0638 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.42 & 0.18 & 0 & 0 & 0 & 0 & 0.28 & 0.12 & 0 \\ 0 & 0 & 0 & 0 & 0.414 & 0.486 & 0 & 0 & 0 & 0 & 0.046 & 0.054 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{P}_3 = \begin{array}{c} \\ (3,1) \\ (3,2) \\ (3,3) \\ (3,4) \\ (3,5) \\ (3,6) \end{array} \begin{array}{c} \begin{array}{cccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (3,6) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) & (4,6) \end{array} \\ \begin{pmatrix} 0.4079 & 0.2242 & 0 & 0 & 0 & 0 & 0.2374 & 0.1305 & 0 & 0 & 0 & 0 \\ 0 & 0.4997 & 0.2903 & 0 & 0 & 0 & 0 & 0.1328 & 0.0772 & 0 & 0 & 0 \\ 0 & 0 & 0.5121 & 0.1426 & 0 & 0 & 0 & 0 & 0.2701 & 0.0752 & 0 & 0 \\ 0 & 0 & 0 & 0.34 & 0.06 & 0 & 0 & 0 & 0 & 0.51 & 0.09 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{P}_4 = \begin{array}{c} \\ (4,1) \\ (4,2) \\ (4,3) \\ (4,4) \\ (4,5) \\ (4,6) \end{array} \begin{array}{c} \begin{array}{cccccccccccc} (4,1) & (4,2) & (4,3) & (4,4) & (4,5) & (4,6) & (5,1) & (5,2) & (5,3) & (5,4) & (5,5) & (5,6) \end{array} \\ \begin{pmatrix} 0.3183 & 0.6318 & 0 & 0 & 0 & 0 & 0.0168 & 0.033 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.375 & 0.125 & 0 & 0 & 0 & 0 & 0.375 & 0.125 & 0 \\ 0 & 0 & 0 & 0 & 0.4037 & 0.1964 & 0 & 0 & 0 & 0 & 0.2691 & 0.1309 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

# Appendix E. Discrete-Time Markov Chain



# Appendix F. Transient State Matrix Partitions

$$\mathbf{Q}_1 = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (1,4) \\ (1,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) \end{array} \\ \begin{pmatrix} 0.352 & 0.448 & 0 & 0 & 0 & 0.088 & 0.112 & 0 & 0 & 0 \\ 0 & 0.48 & 0.32 & 0 & 0 & 0 & 0.12 & 0.08 & 0 & 0 \\ 0 & 0 & 0.608 & 0.192 & 0 & 0 & 0 & 0.152 & 0.048 & 0 \\ 0 & 0 & 0 & 0.34 & 0.06 & 0 & 0 & 0 & 0.51 & 0.09 \\ 0 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{Q}_2 = \begin{array}{c} \\ (2,1) \\ (2,2) \\ (2,3) \\ (2,4) \\ (2,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (2,1) & (2,2) & (2,3) & (2,4) & (2,5) & (3,1) & (3,2) & (3,3) & (3,4) & (3,5) \end{array} \\ \begin{pmatrix} 0.357 & 0.343 & 0 & 0 & 0 & 0.153 & 0.147 & 0 & 0 & 0 \\ 0 & 0.4888 & 0.3613 & 0 & 0 & 0 & 0.0863 & 0.0638 & 0 & 0 \\ 0 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.42 & 0.18 & 0 & 0 & 0 & 0.28 & 0.12 \\ 0 & 0 & 0 & 0 & 0.414 & 0 & 0 & 0 & 0 & 0.046 \end{pmatrix} \end{array}$$

$$\mathbf{Q}_3 = \begin{array}{c} \\ (3,1) \\ (3,2) \\ (3,3) \\ (3,4) \\ (3,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0.4079 & 0.2242 & 0 & 0 & 0 & 0.2374 & 0.1305 & 0 & 0 & 0 \\ 0 & 0.4997 & 0.2903 & 0 & 0 & 0 & 0.1328 & 0.0772 & 0 & 0 \\ 0 & 0 & 0.5121 & 0.1426 & 0 & 0 & 0 & 0.2701 & 0.0752 & 0 \\ 0 & 0 & 0 & 0.34 & 0.06 & 0 & 0 & 0 & 0.51 & 0.09 \\ 0 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{Q}_4 = \begin{array}{c} \\ (4,1) \\ (4,2) \\ (4,3) \\ (4,4) \\ (4,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0.3183 & 0.6318 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.6 & 0.4 & 0 \\ 0 & 0 & 0 & 0.375 & 0.125 \\ 0 & 0 & 0 & 0 & 0.4037 \end{pmatrix} \end{array}$$

# Appendix G. Mean Time in Transient State Matrix Partitions

$$\mathbf{M}_{1,1} = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (1,4) \\ (1,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) \end{array} \\ \begin{pmatrix} 1.5432 & 1.3295 & 1.0853 & 0.3157 & 0.0316 & 0.2112 & 0.7918 & 1.8579 & 1.3285 & 0.4565 \\ 0 & 1.9231 & 1.5699 & 0.4567 & 0.0457 & 0 & 0.4514 & 1.8518 & 1.4893 & 0.5276 \\ 0 & 0 & 2.551 & 0.7421 & 0.0742 & 0 & 0 & 1.2925 & 1.5322 & 0.5846 \\ 0 & 0 & 0 & 1.5152 & 0.1515 & 0 & 0 & 0 & 1.3323 & 0.6419 \\ 0 & 0 & 0 & 0 & 1.6667 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{M}_{1,2} = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (1,4) \\ (1,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0.0546 & 0.223 & 0.2361 & 0.6146 & 0.3622 & 0.019 & 0.0975 & 0.3244 & 0.7375 & 0.2474 \\ 0 & 0.0778 & 0.1053 & 0.6546 & 0.4038 & 0 & 0.0207 & 0.1119 & 0.6184 & 0.2284 \\ 0 & 0 & 0 & 0.65 & 0.4163 & 0 & 0 & 0 & 0.5304 & 0.2093 \\ 0 & 0 & 0 & 0.5652 & 0.3722 & 0 & 0 & 0 & 0.4612 & 0.182 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{M}_{2,1} = \begin{array}{c} \\ (2,1) \\ (2,2) \\ (2,3) \\ (2,4) \\ (2,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1.5552 & 1.0434 & 1.2564 & 0.6499 & 0.1996 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.956 & 2.3553 & 1.2183 & 0.3745 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.3333 & 1.7241 & 0.5296 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.7241 & 0.5296 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.7065 \end{pmatrix} \end{array}$$

$$\mathbf{M}_{2,2} = \begin{array}{c} \\ (2,1) \\ (2,2) \\ (2,3) \\ (2,4) \\ (2,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0.4018 & 0.817 & 0.6223 & 0.4102 & 0.1863 & 0.1399 & 0.4988 & 1.2013 & 1.1784 & 0.3089 \\ 0 & 0.3372 & 0.4562 & 0.6154 & 0.3339 & 0 & 0.0896 & 0.4851 & 0.8675 & 0.2747 \\ 0 & 0 & 0 & 0.7315 & 0.4586 & 0 & 0 & 0 & 0.5969 & 0.2355 \\ 0 & 0 & 0 & 0.7315 & 0.4586 & 0 & 0 & 0 & 0.5969 & 0.2355 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{M}_{3,2} = \begin{array}{c} \\ (3,1) \\ (3,2) \\ (3,3) \\ (3,4) \\ (3,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 1.6888 & 0.7569 & 0.4503 & 0.0973 & 0.0097 & 0.5881 & 1.385 & 2.1813 & 1.5296 & 0.3353 \\ 0 & 1.9989 & 1.1891 & 0.257 & 0.0257 & 0 & 0.5311 & 1.8524 & 1.5383 & 0.3612 \\ 0 & 0 & 2.0494 & 0.4428 & 0.0443 & 0 & 0 & 1.3839 & 1.4937 & 0.3799 \\ 0 & 0 & 0 & 1.5152 & 0.1515 & 0 & 0 & 0 & 1.2364 & 0.4878 \\ 0 & 0 & 0 & 0 & 1.6667 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{M}_{4,2} = \begin{array}{c} \\ (4,1) \\ (4,2) \\ (4,3) \\ (4,4) \\ (4,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1.4668 & 1.8533 & 2.3166 & 1.4827 & 0.3108 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2.5 & 1.6 & 0.3354 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.5 & 1.6 & 0.3354 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.6 & 0.3354 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.6769 \end{pmatrix} \end{array}$$

# Appendix H. Probability of Entering State Matrix Partitions

$$\mathbf{F}_{1,1} = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (1,4) \\ (1,5) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) \end{array} \\ \begin{pmatrix} 1 & 0.6914 & 0.4255 & 0.2084 & 0.0189 & 0.1358 & 0.4048 & 0.5574 & 0.7705 & 0.2675 \\ 0 & 1 & 0.6154 & 0.3014 & 0.0274 & 0 & 0.2308 & 0.5555 & 0.8638 & 0.3092 \\ 0 & 0 & 1 & 0.4898 & 0.0445 & 0 & 0 & 0.3878 & 0.8887 & 0.3426 \\ 0 & 0 & 0 & 1 & 0.0909 & 0 & 0 & 0 & 0.7727 & 0.3762 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{F}_{1,2} = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (1,4) \\ (1,5) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0.0323 & 0.1116 & 0.1152 & 0.4056 & 0.2173 & 0.013 & 0.0488 & 0.1297 & 0.461 & 0.1475 \\ 0 & 0.0389 & 0.0514 & 0.432 & 0.2423 & 0 & 0.0103 & 0.0448 & 0.3865 & 0.1362 \\ 0 & 0 & 0 & 0.429 & 0.2498 & 0 & 0 & 0 & 0.3315 & 0.1248 \\ 0 & 0 & 0 & 0.373 & 0.2233 & 0 & 0 & 0 & 0.2883 & 0.1085 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{F}_{2,1} = \begin{array}{c} \\ (2,1) \\ (2,2) \\ (2,3) \\ (2,4) \\ (2,5) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0.5334 & 0.3769 & 0.3769 & 0.117 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.7066 & 0.7066 & 0.2193 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0.3103 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.3103 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\mathbf{F}_{2,2} = \begin{array}{c} \\ (2,1) \\ (2,2) \\ (2,3) \\ (2,4) \\ (2,5) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0.2379 & 0.4087 & 0.3037 & 0.2707 & 0.1118 & 0.0954 & 0.2494 & 0.4805 & 0.7365 & 0.1842 \\ 0 & 0.1687 & 0.2226 & 0.4062 & 0.2003 & 0 & 0.0448 & 0.194 & 0.5422 & 0.1638 \\ 0 & 0 & 0 & 0.4828 & 0.2751 & 0 & 0 & 0 & 0.373 & 0.1404 \\ 0 & 0 & 0 & 0.4828 & 0.2751 & 0 & 0 & 0 & 0.373 & 0.1404 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{F}_{3,2} = \begin{array}{c} \\ (3,1) \\ (3,2) \\ (3,3) \\ (3,4) \\ (3,5) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 1 & 0.3787 & 0.2197 & 0.0642 & 0.0058 & 0.4009 & 0.6925 & 0.8725 & 0.956 & 0.2 \\ 0 & 1 & 0.5802 & 0.1696 & 0.0154 & 0 & 0.2655 & 0.741 & 0.9615 & 0.2154 \\ 0 & 0 & 1 & 0.2923 & 0.0266 & 0 & 0 & 0.5535 & 0.9336 & 0.2266 \\ 0 & 0 & 0 & 1 & 0.0909 & 0 & 0 & 0 & 0.7727 & 0.2909 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{F}_{4,2} = \begin{array}{c} \\ (4,1) \\ (4,2) \\ (4,3) \\ (4,4) \\ (4,5) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0.9267 & 0.9267 & 0.9267 & 0.1853 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

# Appendix I. Damage Cost Per State Matrix Partitions

$$\mathbf{C}_{1,1} = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (1,4) \\ (1,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) \end{array} \\ \begin{pmatrix} 1.2346 & 4.2545 & 5.2096 & 0.3789 & 0.0474 & 0.1478 & 3.0288 & 11.1477 & 2.3912 & 0.6163 \\ 0 & 6.1538 & 7.5353 & 0.548 & 0.0685 & 0 & 1.7265 & 11.1105 & 2.6807 & 0.7123 \\ 0 & 0 & 12.2449 & 0.8905 & 0.1113 & 0 & 0 & 7.7551 & 2.758 & 0.7892 \\ 0 & 0 & 0 & 1.8182 & 0.2273 & 0 & 0 & 0 & 2.3981 & 0.8666 \\ 0 & 0 & 0 & 0 & 2.5 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{C}_{1,2} = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (1,4) \\ (1,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0.0983 & 1.2953 & 0.8282 & 0.7375 & 0.5432 & 0.0181 & 0.4876 & 0.9731 & 1.7701 & 0.2226 \\ 0 & 0.4519 & 0.3692 & 0.7855 & 0.6056 & 0 & 0.1034 & 0.3358 & 1.4843 & 0.2056 \\ 0 & 0 & 0 & 0.78 & 0.6244 & 0 & 0 & 0 & 1.273 & 0.1884 \\ 0 & 0 & 0 & 0.6783 & 0.5583 & 0 & 0 & 0 & 1.1069 & 0.1638 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{C}_{2,1} = \begin{array}{c} \\ (2,1) \\ (2,2) \\ (2,3) \\ (2,4) \\ (2,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (2,1) & (2,2) & (2,3) & (2,4) & (2,5) \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1.0886 & 3.991 & 7.5385 & 1.1698 & 0.2695 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7.4817 & 14.132 & 2.1929 & 0.5052 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 3.1034 & 0.715 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.1034 & 0.715 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.3038 \end{pmatrix} \end{array}$$

$$\mathbf{C}_{2,2} = \begin{array}{c} \\ (2,1) \\ (2,2) \\ (2,3) \\ (2,4) \\ (2,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0.7239 & 4.7446 & 2.1826 & 0.4922 & 0.2794 & 0.1329 & 2.4938 & 3.6038 & 2.8282 & 0.278 \\ 0 & 1.9584 & 1.5998 & 0.7385 & 0.5008 & 0 & 0.448 & 1.4552 & 2.0821 & 0.2472 \\ 0 & 0 & 0 & 0.8777 & 0.6879 & 0 & 0 & 0 & 1.4325 & 0.2119 \\ 0 & 0 & 0 & 0.8777 & 0.6879 & 0 & 0 & 0 & 1.4325 & 0.2119 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{C}_{3,2} = \begin{array}{c} \\ (3,1) \\ (3,2) \\ (3,3) \\ (3,4) \\ (3,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 3.0425 & 4.3957 & 1.5792 & 0.1168 & 0.0146 & 0.5587 & 6.9249 & 6.5438 & 3.671 & 0.3018 \\ 0 & 11.6087 & 4.1705 & 0.3083 & 0.0385 & 0 & 2.6553 & 5.5572 & 3.692 & 0.3251 \\ 0 & 0 & 7.1876 & 0.5314 & 0.0664 & 0 & 0 & 4.1516 & 3.5849 & 0.3419 \\ 0 & 0 & 0 & 1.8182 & 0.2273 & 0 & 0 & 0 & 2.9673 & 0.439 \\ 0 & 0 & 0 & 0 & 2.5 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\mathbf{C}_{4,2} = \begin{array}{c} \\ (4,1) \\ (4,2) \\ (4,3) \\ (4,4) \\ (4,5) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1.3935 & 9.2666 & 6.9499 & 3.5584 & 0.2797 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 7.5 & 3.84 & 0.3018 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7.5 & 3.84 & 0.3018 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.84 & 0.3018 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.5092 \end{pmatrix} \end{array}$$

# Appendix J. MATLAB - Stochastic Game Data

```
%% Blue payoffs per stage game %%
Blue1_1 = [0.2 0.1 0.2 0.1;
           0.1 0.2 0.3 0.05];
BFail1_1 = 1-Blue1_1;

Blue1_2 = [0.2 0.3 0 0 0.3;
           0.2 0.3 0 0.2 0.3];
BFail1_2 = 1 - Blue1_2;

Blue1_3 = [0.2 0.5 0.7 0.4 0.6;
           0.2 0.4 0 0.3 0.4];
BFail1_3 = 1 - Blue1_3;

Blue1_4 = [0.6 0.6 0.7 0.5;
           0.3 0.3 0.4 0.3];
BFail1_4 = 1 - Blue1_4;

Blue1_5 = [0 0 0;
           0 0 0];
BFail1_5 = 1 - Blue1_5;

Blue2_1 = [0 0 0 0.1;
           0.3 0.1 0.2 0;
           0 0 0 0];

% %Updated from IP
% Blue2_1 = [0 0 0 0.1;
%            0.2 0.1 0.1 0;
%            0 0 0 0];
% %%
```

```
31
   BFail2_1 = 1 - Blue2_1;
33
   Blue2_2 = [0 0.1  0.4  0.4  0;
35             0  0.2  0.2  0.1  0.3;
               0.2  0  0  0.2  0.1];
37
   %Updated from IP
39 % Blue2_2 = [0 0.1 0.4  0.4  0;
   %              0  0.3  0.3  0.3  0.2;
41 %              0.4  0  0  0.3  0.4];
   % %%
43
   BFail2_2 = 1 - Blue2_2;
45
   Blue2_3 = [0 0 0 0  0;
47            0  0.2  0.8  0.7  0.2;
              0  0.4  0.5  0.4  0.7];
49
   %%Updated from IP
51 % Blue2_3 = [0 0 0 0 0;
   %              0  0.1  0.9  0.9  0.4;
53 %              0  0.4  0.6  0.3  0.6];
   % %%
55
   BFail2_3 = 1 - Blue2_3;
57
   Blue2_4 = [0 0.6  0  0.1;
59            0.1  0  0  0.4;
              0.5  0.4  0.4  0.5];
61
   %%Updated from IP
```

```
% Blue2_4 = [0 0.6 0 0.1;
%           0.1 0 0 0.3;
%           0.5 0.4 0.5 0.5];
% %%

BFail2_4 = 1 - Blue2_4;

Blue2_5 = [0.1 0.1 0.1;
           0 0 0;
           0 0 0];

%%Updated from IP
% Blue2_5 = [0.1 0.1 0.1;
%           0 0 0;
%           0 0 0];
% %%

BFail2_5 = 1 - Blue2_5;

Blue3_1 = [0.3 0.4 0.5 0.2;
           0.5 0.7 0.7 0.1;
           0 0 0 0.6;
           0.4 0.3 0.7 0.3;
           0 0 0 0];
BFail3_1 = 1 - Blue3_1;

Blue3_2 = [0.2 0.4 0 0.3 0.7;
           0 0.6 0.1 0.1 0.6;
           0 0 0.7 0 0;
           0.1 0.5 0 0.1 0.2;
           0.3 0 0 0 0.7];
BFail3_2 = 1 - Blue3_2;
```

```
95
    Blue3_3 = [0 0.1  0.4  0.1  0.3;
97              0.4  0.5  0.3  0.3  0.3;
                0.1  0  0  0  0;
99              0.3  0  0.6  0.4  0.3;
                0  0.8  0.2  0.6  0.8];
101 BFail3_3 = 1 - Blue3_3;

103 Blue3_4 = [0.4  0  0  0;
                0.4  0.2  0.2  0;
105             0  0.2  0  0;
                0.2  0.6  0.6  0.5;
107             0.7  0.8  0.7  0.5];
    BFail3_4 = 1 - Blue3_4;
109
    Blue3_5 = [0 0  0;
111             0.2  0  0;
                0  0  0;
113             0  0  0;
                0  0  0];
115 BFail3_5 = 1 - Blue3_5;

117 Blue4_1 = [0.05 0  0  0;
                0  0  0  0;
119             0  0  0  0;
                0  0  0  0];
121
    %%Updated from IP
123 % Blue4_1 = [0.05 0 0 0;
    %               0 0 0 0;
125 %               0 0 0 0;
    %               0 0 0 0];
```

```
% %%

BFail4_1 = 1 - Blue4_1;

Blue4_2 = [0 0.1  0.5  0.2  0;
           0.4  0  0.4  0  0;
           0  0  0  0  0;
           0  0  0  0  0];

%%Updated from IP
% Blue4_2 = [0 0.1 0.5  0.2  0;
%            0.5  0  0.4  0  0;
%            0  0  0  0  0;
%            0  0  0  0  0];
% %%

BFail4_2 = 1 - Blue4_2;

Blue4_3 = [0.5  0.1  0  0.2  0;
           0  0  0  0.3  0;
           0  0  0  0.2  0;
           0  0  0  0.5  0];

%%Updated from IP
% Blue4_3 = [0.5 0.1 0  0.2  0;
%            0  0  0  0.4  0;
%            0  0  0  0.3  0;
%            0  0  0  0.4  0];
% %%

BFail4_3 = 1 - Blue4_3;
```

```
Blue4_4 = [0.1  0.1  0  0.2;
           0  0.1  0.5  0.5;
           0 0 0 0;
           0 0 0  0];

%Updated from IP
% Blue4_4 = [0.1 0.1 0  0.2;
%            0  0.2  0.5  0.6;
%            0 0 0 0;
%            0 0 0  0];
% %%

BFail4_4 = 1 - Blue4_4;

Blue4_5 = [0.5  0.3  0.3;
           0.1  0.1  0.1;
           0.3  0.7  0.1;
           0.2  0.2  0.6];

%%Updated from IP
% Blue4_5 = [0.5 0.3  0.3;
%            0.1  0.3  0.2;
%            0.2  0.8  0.3;
%            0.1  0.1  0.8];
% %%

BFail4_5 = 1 - Blue4_5;

BluePayoffs = {Blue1_1 Blue1_2 Blue1_3 Blue1_4 Blue1_5;
               Blue2_1 Blue2_2 Blue2_3 Blue2_4 Blue2_5;
               Blue3_1 Blue3_2 Blue3_3 Blue3_4 Blue3_5;
               Blue4_1 Blue4_2 Blue4_3 Blue4_4 Blue4_5};
```

```
191
    BlueFail = {BFail1_1 BFail1_2 BFail1_3 BFail1_4 BFail1_5;
193                 BFail2_1 BFail2_2 BFail2_3 BFail2_4 BFail2_5;
                    BFail3_1 BFail3_2 BFail3_3 BFail3_4 BFail3_5;
195                 BFail4_1 BFail4_2 BFail4_3 BFail3_4 BFail4_5};

197 %% Red baseline probabilities of success to establish payoffs %%
    RedBaseline1 = [0.7 0.3 0.5  0.4];
199 RedBaseline2 = [0.5 0.3 0.4  0.2  0.5];
    RedBaseline3 = [0.3 0.2 0.3  0.2  0.4];
201 RedBaseline4 = [0.1 0.3 0.5  0.3];
    RedBaseline5 = [0.2 0.5  0.6];
203
    RedBaseline = {RedBaseline1 RedBaseline2 RedBaseline3 RedBaseline4 RedBaseline5};
205
    %% Red payoffs using Blue payoffs and Red baseline probabilities %%
207 RedPayoffs = {};
    for i = 1 : size(BluePayoffs,1)
209     for j = 1 : size(BluePayoffs, 2)
            tempBlue = BluePayoffs{i,j};
211         [n,m] = size(tempBlue);
            tempRedBaseline = RedBaseline{1,j};
213         tempRedPayoff = zeros(n,m);
            for ii = 1 : n
215             for jj = 1 : m
                    tempRedPayoff(ii,jj) = tempRedBaseline(:,jj)*(1 - tempBlue(ii,jj));
217             end
            end
219         RedPayoffs(i,j) = {tempRedPayoff};
        end
221 end
```

```
%% Set-up for DTMC %%
[n,m] = size(BluePayoffs);
BMNE = sym('x', [n+1 m+1]); %Blue Mixed Nash Equilibrium
RMNE = sym('y', [n+1 m+1]); %Red Mixed Nash Equilibrium

x1_1 = BluePayoffs{1,1}(1,1);
y1_1 = RedPayoffs{1,1}(1,1);

x1_2 = 0.5*BluePayoffs{1,2}(1,1) + 0.5*BluePayoffs{1,2}(2,1);
y1_2 = 0.5*RedPayoffs{1,2}(1,1) + 0.5*RedPayoffs{1,2}(2,1);

x1_3 = 0.286*BluePayoffs{1,3}(1,1) + 0.714*BluePayoffs{1,3}(2,1) ;
y1_3 = 0.286*RedPayoffs{1,3}(1,1) + 0.714*RedPayoffs{1,3}(2,1);

x1_4 = 0.5*BluePayoffs{1,4}(1,3) + 0.5*BluePayoffs{1,4}(1,4) ;
y1_4 = 0.5*RedPayoffs{1,4}(1,3) + 0.5*RedPayoffs{1,4}(1,4);

x1_5 = 0.5*BluePayoffs{1,5}(1,3) + 0.5*BluePayoffs{1,5}(2,3) ;
y1_5 = 0.5*RedPayoffs{1,5}(1,3) + 0.5*RedPayoffs{1,5}(2,3);

x1_6 = 0;
y1_6 = 1;

x2_1 = BluePayoffs{2,1}(2,1) ;
y2_1 = RedPayoffs{2,1}(2,1);

x2_2 = 0.125*BluePayoffs{2,2}(2,1) + 0.125*BluePayoffs{2,2}(2,5) + 0.375*BluePayoffs{2,2}(3,1)
    + 0.375*BluePayoffs{2,2}(3,5) ;
y2_2 = 0.125*RedPayoffs{2,2}(2,1) + 0.125*RedPayoffs{2,2}(2,5) + 0.375*RedPayoffs{2,2}(3,1) +
    0.375*RedPayoffs{2,2}(3,5);

x2_3 = 0.9*BluePayoffs{2,3}(2,1) + 0.1*BluePayoffs{2,3}(3,1);
```

```
y2_3 = 0.9*RedPayoffs{2,3}(2,1) + 0.1*RedPayoffs{2,3}(3,1);

x2_4 = BluePayoffs{2,4}(3,3);
y2_4 = RedPayoffs{2,4}(3,3);

x2_5 = BluePayoffs{2,5}(1,3);
y2_5 = RedPayoffs{2,5}(1,3);

x2_6 = 0;
y2_6 = 1;

x3_1 = 0.6231*BluePayoffs{3,1}(2,1) + 0.3069*BluePayoffs{3,1}(2,4) + 0.0469*BluePayoffs
    {3,1}(4,1) + 0.0231*BluePayoffs{3,1}(4,4);
y3_1 = 0.6231*RedPayoffs{3,1}(2,1) + 0.3069*RedPayoffs{3,1}(2,4) + 0.0469*RedPayoffs
    {3,1}(4,1) + 0.0231*RedPayoffs{3,1}(4,4);

x3_2 = 0.0812*BluePayoffs{3,2}(3,1) + 0.0348*BluePayoffs{3,2}(3,3) + 0.6188*BluePayoffs
    {3,2}(5,1) + 0.2652*BluePayoffs{3,2}(5,3);
y3_2 = 0.0812*RedPayoffs{3,2}(3,1) + 0.0348*RedPayoffs{3,2}(3,3) + 0.6188*RedPayoffs
    {3,2}(5,1) + 0.2652*RedPayoffs{3,2}(5,3);

x3_3 = 0.3071*BluePayoffs{3,3}(2,1) + 0.1026*BluePayoffs{3,3}(2,3) + 0.2652*BluePayoffs
    {3,3}(2,5) + 0.0068*BluePayoffs{3,3}(4,1) + 0.0023*BluePayoffs{3,3}(4,3) + 0.0059*
    BluePayoffs{3,3}(4,5) + 0.1411*BluePayoffs{3,3}(5,1) + 0.0472*BluePayoffs{3,3}(5,3) +
    0.1219*BluePayoffs{3,3}(5,5);
y3_3 = 0.3071*RedPayoffs{3,3}(2,1) + 0.1026*RedPayoffs{3,3}(2,3) + 0.2652*RedPayoffs
    {3,3}(2,5) + 0.0068*RedPayoffs{3,3}(4,1) + 0.0023*RedPayoffs{3,3}(4,3) + 0.0059*
    RedPayoffs{3,3}(4,5) + 0.1411*RedPayoffs{3,3}(5,1) + 0.0472*RedPayoffs{3,3}(5,3) +
    0.1219*RedPayoffs{3,3}(5,5);

x3_4 = 0.5*BluePayoffs{3,4}(5,3) + 0.5*BluePayoffs{3,4}(5,4);
y3_4 = 0.5*RedPayoffs{3,4}(5,3) + 0.5*RedPayoffs{3,4}(5,4);
```

```
275
    x3_5 = 0.2*BluePayoffs{3,5}(1,3) + 0.2*BluePayoffs{3,5}(2,3) + 0.2*BluePayoffs{3,5}(3,3) + 0.2*
        BluePayoffs{3,5}(4,3) + 0.2*BluePayoffs{3,5}(5,3);
277 y3_5 = 0.2*RedPayoffs{3,5}(1,3) + 0.2*RedPayoffs{3,5}(2,3) + 0.2*RedPayoffs{3,5}(3,3) + 0.2*
        RedPayoffs{3,5}(4,3) + 0.2*RedPayoffs{3,5}(5,3);

279 x3_6 = 0;
    y3_6 = 1;
281
    x4_1 = BluePayoffs{4,1}(1,1) ;
283 y4_1 = RedPayoffs{4,1}(1,1);

285 x4_2 = 0.25*BluePayoffs{4,2}(1,5) + 0.25*BluePayoffs{4,2}(2,5) + 0.25*BluePayoffs{4,2}(3,5) +
        0.25*BluePayoffs{4,2}(4,5);
    y4_2 = 0.25*RedPayoffs{4,2}(1,5) + 0.25*RedPayoffs{4,2}(2,5) + 0.25*RedPayoffs{4,2}(3,5) +
        0.25*RedPayoffs{4,2}(4,5);
287
    x4_3 = 0.25*BluePayoffs{4,3}(1,5) + 0.25*BluePayoffs{4,3}(2,5) + 0.25*BluePayoffs{4,3}(3,5) +
        0.25*BluePayoffs{4,3}(4,5);
289 y4_3 = 0.25*RedPayoffs{4,3}(1,5) + 0.25*RedPayoffs{4,3}(2,5) + 0.25*RedPayoffs{4,3}(3,5) +
        0.25*RedPayoffs{4,3}(4,5);

291 x4_4 = BluePayoffs{4,4}(2,3);
    y4_4 = RedPayoffs{4,4}(2,3);
293
    x4_5 = 0.145*BluePayoffs{4,5}(3,2) + 0.145*BluePayoffs{4,5}(3,3) + 0.355*BluePayoffs{4,5}(4,2)
        + 0.355*BluePayoffs{4,5}(4,3);
295 y4_5 = 0.145*RedPayoffs{4,5}(3,2) + 0.145*RedPayoffs{4,5}(3,3) + 0.355*RedPayoffs{4,5}(4,2) +
         0.355*RedPayoffs{4,5}(4,3);

297 x4_6 = 0;
    y4_6 = 0;
```

```
299
    x5_1 = 1;
301 y5_1 = 0;

303 x5_2 = 1;
    y5_2 = 0;
305
    x5_3 = 1;
307 y5_3 = 0;

309 x5_4 = 1;
    y5_4 = 0;
311
    x5_5 = 1;
313 y5_5 = 0;

315 x5_6 = 1;
    y5_6 = 0;
317
    %% Cost to Blue where first value is O&M,                %%
319 %% second is attack damage cost times probability of    %%
    %% Blue failure. Values in thousands.                   %%
321 c1_1 = 1*(BlueFail{1,1}(1,1));
    c1_2 = 0.5*(4*BlueFail{1,2}(1,1)) + 0.5*(4*BlueFail{1,2}(2,1));
323 c1_3 = 0.286*(6*BlueFail{1,3}(1,1)) + 0.714*(6*BlueFail{1,3}(2,1));
    c1_4 = 0.5*(3*BlueFail{1,4}(1,3))+0.5*(3*BlueFail{1,4}(1,4));
325 c1_5 = 0.5*(1.5*BlueFail{1,5}(1,3)) + 0.5*(1.5*BlueFail{1,5}(2,3));

327 c2_1 = 1*BlueFail{2,1}(2,1);
    c2_2 = 0.125*(4*BlueFail{2,2}(2,1))+0.125*(5*BlueFail{2,2}(2,5))+0.375*(4*BlueFail{2,2}(3,1))
        +0.375*(5*BlueFail{2,2}(3,5));
329 c2_3 = 0.9*(6*BlueFail{2,3}(2,1))+0.1*(6*BlueFail{2,3}(3,1));
```

```
c2_4 = 3*BlueFail{2,4}(3,3);
c2_5 = 1.5*BlueFail{2,5}(1,3);

c3_1 = 0.6231*(1*BlueFail{3,1}(2,1))+0.3069*(5*BlueFail{3,1}(2,4))+0.0469*(1*BlueFail
    {3,1}(4,1))+0.0231*(5*BlueFail{3,1}(4,4));
c3_2 = 0.812*(4*BlueFail{3,2}(3,1))+0.0348*(3*BlueFail{3,2}(3,3))+0.6188*(4*BlueFail{3,2}(5,1)
    )+0.2652*(3*BlueFail{3,2}(5,3));
c3_3 = 0.3071*(6*BlueFail{3,3}(2,1))+0.1026*(8*BlueFail{3,3}(2,3))+0.2652*(3*BlueFail
    {3,3}(2,5))+0.0068*(6*BlueFail{3,3}(4,1))+0.0023*(8*BlueFail{3,3}(4,3))+0.0059*(3*BlueFail
    {3,3}(4,5))+0.1411*(6*BlueFail{3,3}(5,1))+0.0472*(8*BlueFail{3,3}(5,3))+0.1219*(3*BlueFail
    {3,3}(5,5));
c3_4 = 0.5*(3*BlueFail{3,4}(5,3))+0.5*(3*BlueFail{3,4}(5,4));
c3_5 = 0.2*(1.5*BlueFail{3,5}(1,3))+0.2*(1.5*BlueFail{3,5}(2,3))+0.2*(1.5*BlueFail{3,5}(3,3))
    +0.2*(1.5*BlueFail{3,5}(4,3))+0.2*(1.5*BlueFail{3,5}(5,3));

c4_1 = 1*BlueFail{4,1}(1,1);
c4_2 = 0.25*(5*BlueFail{4,2}(1,5))+0.25*(5*BlueFail{4,2}(2,5))+0.25*(5*BlueFail{4,2}(3,5))
    +0.25*(5*BlueFail{4,2}(4,5));
c4_3 = 0.25*(3*BlueFail{4,3}(1,5))+0.25*(3*BlueFail{4,3}(2,5))+0.25*(3*BlueFail{4,3}(3,5))
    +0.25*(3*BlueFail{4,3}(4,5));
c4_4 = 3*BlueFail{4,4}(2,3);
c4_5 = 0.145*(1.5*BlueFail{4,5}(3,2))+0.145*(1.5*BlueFail{4,5}(3,3))+0.355*(1.5*BlueFail
    {4,5}(4,2))+0.355*(1.5*BlueFail{4,5}(4,3));

c = [c1_1 c1_2 c1_3 c1_4 c1_5 c2_1 c2_2 c2_3 c2_4 c2_5 c3_1 c3_2 c3_3 c3_4 c3_5 c4_1 c4_2 c4_3
    c4_4 c4_5 ]';
```

**MATLAB - Stochastic Game Data**

# Appendix K. MATLAB - Stochastic Game Model

```
%% Stochastic model assuming NE is maintained at each stage game no matter
%% the values assigned to actions.

%% Pull in game data from StochGameData.m file
StochGameData;

%% Create transition matrix symbolically
[n,m] = size(BluePayoffs);
n = n+1;
m = m+1;
P = sym(zeros(n*m, n*m));
for i = 1 : size(BMNE,1)
    for j = 1 : size(BMNE,2)-1
        begin = (i-1)*m;
        if begin+j < 24
            P(begin+j, begin+j+1) = (1 - BMNE(i,j)) * RMNE(i,j);
            P(begin+j, begin+j+m) = BMNE(i,j) * (1-RMNE(i,j));
            P(begin+j, begin+j+m+1) = BMNE(i,j) * RMNE(i,j);
        else
        end
    end
end

for i = 1 : size(P,1)
    P(i,i) = 1 - sum(P(i,:));
end

%% Evaluate transition probabilities by substituting variables with values assigned in
    StochGameData.m
P = eval(subs(P));
```

```
%% Initial state vector, initialize stationary proportions
PI(1,:) = [1,zeros(1,29)];

%% Set number of runs
runCount = 40;
for i = 2:runCount
    %% Iterate to find stationary probabilities
    PI(i,:) = PI(1,:)*(P^i);
end

%% Plot the results
plot(1:runCount,PI)
xlabel('Time Steps')
ylabel('Probability')
title('Stationary Probabilities')

%% Find transient state matrix
Q = [P(1:5,1:5),P(1:5,7:11), P(1:5,13:17), P(1:5,19:23); P(7:11,1:5),P(7:11,7:11), P(7:11,13:17)
    , P(7:11,19:23); P(13:17,1:5), P(13:17,7:11), P(13:17, 13:17), P(13:17,19:23); P(19:23,1:5)
    , P(19:23,7:11), P(19:23,13:17), P(19:23,19:23)];

%% Find expected total number of time periods spent in state j given starting in state i
M = inv((eye(size(Q)) - Q));

%% Find probability of ever transitioning to state j given starting in state i
F = zeros(size(M));
for i = 1 : size(M,1)
    for j = 1 : size(M,2)
        F(i,j) = M(i,j)/M(j,j);
    end
end
```

```
61 %% Find cost from state i to state j
   indiv (:,:) = zeros(size(M));
63 for i = 1 : size(M,1)
       for j = 1 : size(M,2)
65         indiv(i,j) = M(i,j)*c(j,1);
       end
67 end

69 %% Display results
   P
71 PI(runCount, :)
   Q
73 M
   F
75 indiv

77 %% Estimated total cost given start in state i, based on network architecture
   C = M*c
79 Total = sum(C)
```

**MATLAB - Stochastic Game Model**

# Appendix L. LINGO - Integer Program Formulation

```
SETS:
RED_ACTIONS/1..21/: D;
BLUE_ACTIONS/1..14/: COST_REPAIR, COST_UPGRADE, X, Y, Z, TOTAL_REPAIR, TOTAL_UPGRADE, TOTAL_DN;
PROBS(BLUE_ACTIONS,RED_ACTIONS): REPAIR_UTIL, UPGRADE_UTIL, MSNE, DN_UTIL;
ENDSETS

DATA:
REPAIR_UTIL =  0.2  0.1 0.2 0.1  0.2 0.3 0   0   0.3 0.2  0.5 0.7 0.4 0.6 0.6 0.6 0.7 0.5 0   0   0
 0.1  0.2 0.3 0.05 0.2 0.3 0   0.2 0.3 0.2  0.4 0   0.3 0.4 0.3 0.3 0.4 0.3 0   0   0
 0    0   0   0.1  0   0.1 0.4 0.4 0   0    0   0   0   0   0   0.6 0   0.1 0.1 0.1 0.1
 0.3  0.1 0.2 0    0   0.2 0.2 0.1 0.3 0    0.2 0.8 0.7 0.2 0.1 0   0   0.4 0   0   0
 0    0   0   0    0.2 0   0   0.2 0.1 0    0.4 0.5 0.4 0.7 0.5 0.4 0.4 0.5 0   0   0
 0.3  0.4 0.5 0.2  0.2 0.4 0   0.3 0.7 0    0.1 0.4 0.1 0.3 0.4 0   0   0   0   0   0
 0.5  0.7 0.7 0.1  0   0.6 0.1 0.1 0.6 0.4  0.5 0.3 0.3 0.3 0.4 0.2 0.2 0   0.2 0   0
 0    0   0   0.6  0   0   0.7 0   0   0.1  0   0   0   0   0   0.2 0   0   0   0   0
 0.4  0.3 0.7 0.3  0.1 0.5 0   0.1 0.2 0.3  0   0.6 0.4 0.3 0.2 0.6 0.6 0.5 0   0   0
 0    0   0   0    0.3 0   0   0   0.7 0    0.8 0.2 0.6 0.8 0.7 0.8 0.7 0.5 0   0   0
 0.05 0   0   0    0   0.1 0.5 0.2 0   0.05 0.1 0   0.2 0   0.1 0.1 0   0.2 0.5 0.3 0.3
 0    0   0   0    0.4 0   0.4 0   0   0    0   0   0.3 0   0   0.1 0.5 0.5 0.1 0.1 0.1
 0    0   0   0    0   0   0   0   0   0    0   0   0.2 0   0   0   0   0   0.3 0.7 0.1
 0    0   0   0    0   0   0   0   0   0    0   0   0.5 0   0   0   0   0   0.1 0.2 0.6 ;

UPGRADE_UTIL =  0.4 0.3 0.4 0.1  0.1 0.5 0.1 0   0   0.4  0.1 0.6 0.6 0.5 0.6 0.6 0.7 0.6 0   0   0
  0.2 0.2 0.2 0.1  0.1 0.4 0   0.3 0.2 0.2  0.4 0   0.2 0.5 0.4 0.4 0.5 0.3 0   0   0
  0   0   0   0.2  0   0.1 0.3 0.5 0   0    0   0   0   0   0   0.7 0   0.2 0.1 0.3 0.2
  0.2 0.1 0.1 0    0   0.3 0.3 0.3 0.2 0    0.1 0.9 0.9 0.4 0.1 0   0   0.3 0   0   0
  0   0   0   0    0.4 0   0   0.3 0.4 0    0.4 0.6 0.3 0.6 0.5 0.4 0.5 0.5 0   0   0
  0.2 0.4 0.5 0.3  0.4 0.5 0   0.3 0.7 0    0.2 0.3 0.1 0.4 0.4 0   0   0   0   0   0
  0.4 0.7 0.8 0.2  0   0.5 0.1 0.1 0.5 0.5  0.5 0.4 0.2 0.4 0.5 0.4 0.2 0   0.2 0   0
  0   0   0   0.7  0   0   0.6 0   0   0.2  0   0   0   0   0   0.4 0   0   0   0   0
  0.3 0.4 0.8 0.3  0.2 0.5 0   0.2 0.2 0.2  0   0.5 0.5 0.5 0.1 0.5 0.4 0.5 0   0   0
  0   0   0   0    0.5 0   0   0   0.8 0    0.8 0.4 0.5 0.9 0.6 0.8 0.8 0.4 0   0   0
  0.2 0   0   0    0   0.3 0.4 0.2 0   0.2  0.2 0   0.2 0   0.3 0.2 0   0.3 0.5 0.2 0.4
  0   0   0   0    0.5 0   0.4 0   0   0    0   0   0.4 0   0   0.2 0.5 0.6 0.1 0.3 0.2
  0   0   0   0    0   0   0   0   0   0    0   0   0.3 0   0   0   0   0   0.2 0.8 0.3
  0   0   0   0    0   0   0   0   0   0    0   0   0.4 0   0   0   0   0   0.2 0.1 0.8 ;

D = 1 2 2 5 4 3 3 3.5 5 6 5 8 5 3 2.5 5 3 3 2 1.5 1.5 ;
```

```
MSNE =  1      0 0 0      0.5    0 0      0 0     0.286  0 0      0 0      0 0 0.5 0.5 0 0     0.5
  0      0 0 0      0.5    0 0      0 0     0.714  0 0      0 0      0 0 0   0   0 0     0.5
  0      0 0 0      0      0 0      0 0     0      0 0      0 0      0 0 0   0   0 0     1
  1      0 0 0      0.125  0 0      0 0.125 0.9    0 0      0 0      0 0 0   0   0 0     0
  0      0 0 0      0.375  0 0      0 0.375 0.1    0 0      0 0      0 0 1   0   0 0     0
  0      0 0 0      0      0 0      0 0     0      0 0      0 0      0 0 0   0   0 0     0.2
  0.6231 0 0 0.3069 0      0 0      0 0     0.3071 0 0.1026 0 0.2652 0 0 0   0   0 0     0.2
  0      0 0 0      0.812  0 0.0348 0 0     0      0 0      0 0      0 0 0   0   0 0     0.2
  0.0469 0 0 0.0231 0      0 0      0 0     0.0068 0 0.0023 0 0.0059 0 0 0   0   0 0     0.2
       0      0 0 0      0.6188 0 0.2652 0 0     0.1411 0 0.0472 0 0.1219 0 0 0.5 0.5 0 0     0.2
  1      0 0 0      0      0 0      0 0.25  0      0 0      0 0.25   0 0 0   0   0 0     0
       0      0 0 0      0      0 0      0 0.25  0      0 0      0 0.25   0 0 1   0   0 0     0
       0      0 0 0      0      0 0      0 0.25  0      0 0      0 0.25   0 0 0   0   0 0.145 0.145
       0      0 0 0      0      0 0      0 0.25  0      0 0      0 0.25   0 0 0   0   0 0.355 0.355 ;

COST_REPAIR =  4 2 1 3 4 2.5 5 0.75 1.5 2 1.3 1.5 1 1 ;
COST_UPGRADE = 7 3 2 2 4 6 8 2 8 5 1.5 1 1 1  ;
EFFECT_LVL = 0.8;
ENDDATA

!OBJECTIVE FUNCTION: sum of Blue utilities based on decision made;
MAX = @SUM(PROBS(I,J): REPAIR_UTIL(I,J)*(X(I)+EFFECT_LVL*Z(I))+UPGRADE_UTIL(I,J)*Y(I));

!SUCH THAT:;
!calculate total repair cost per blue action;
@FOR(BLUE_ACTIONS(I) : TOTAL_REPAIR(I)=(COST_REPAIR(I) + @SUM(RED_ACTIONS(J):(1-REPAIR_UTIL(I,J))*MSNE(I,J)*D(J))));
!calculate total upgrade cost per blue action;
@FOR(BLUE_ACTIONS(I) : TOTAL_UPGRADE(I)=(COST_UPGRADE(I) + @SUM(RED_ACTIONS(J):(1-UPGRADE_UTIL(I,J))*MSNE(I,J)*D(J))));
!calculate total cost of no action;
@FOR(BLUE_ACTIONS(I) : TOTAL_DN(I)= @SUM(RED_ACTIONS(J):(1-(EFFECT_LVL*REPAIR_UTIL(I,J)))*MSNE(I,J)*D(J)));
!restrict decision to either repair, upgrade, do nothing for each action;
@FOR(BLUE_ACTIONS(I): X(I)+Y(I)+Z(I) = 1);
!cost threshold;
@SUM(BLUE_ACTIONS(I): TOTAL_REPAIR(I)*X(I) + TOTAL_UPGRADE(I)*Y(I) + TOTAL_DN(I)*Z(I)) <= 70;
!choice variables must be binary;
@FOR(BLUE_ACTIONS(I): @BIN(X(I)));
@FOR(BLUE_ACTIONS(I): @BIN(Y(I)));
!provide output for utilities of no action;
@FOR(PROBS(I,J) : DN_UTIL = EFFECT_LVL * REPAIR_UTIL(I,J));
```

# Bibliography


1. T. Fujiwara-Greve, *Non-Cooperative Game Theory*. Tokyo: Springer, 2015.

2. Y. Y. Haimes, "On the Definition of Resilience in Systems," *Risk Analysis*, vol. 29, no. 4, pp. 498–501, 2009.

3. A. Wilner, "Cyber Deterrence and Critical-Infrastructure Protection: Expectation, Application, and Limitation," *Comparative Strategy*, vol. 36, no. 4, pp. 309–318, 2017.

4. S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A Survey of Game Theory as Applied to Network Security," *2010 43rd Hawaii International Conference on System Sciences*, pp. 1–10, 2010.

5. J. R. Lindsay, "Stuxnet and the Limits of Cyber Warfare," *Security Studies*, vol. 22, no. 3, pp. 365–404, 2013.

6. J. Brown, "Historically Speaking Cyberwar Isn't So New - It Began in 1967," *Army Magazine*, pp. 65–68, 9 2017.

7. M. Warner, "Cybersecurity: A Pre-history," *Intelligence and National Security*, vol. 27, no. 5, pp. 781–799, 2012.

8. W. Gibson, *Neuromancer*. New York: Ace Books, 1984.

9. R. Ottis and P. Lorents, "Cyberspace: Definition and Implications," in *The 5th International Conference on Information Warfare and Security*, (Reading), pp. 267–270, Academic Publishing Limited, 2010.

10. D. Ventre, *Information Warfare*. John Wiley & Sons, 2nd ed., 2016.

11. F. D. Kramer, S. H. Starr, and L. K. Wentz, *Cyberpower and National Security*. Washington, D.C.: National Defense University Press, 2009.

12. M. D. Cavelty, "The Militarisation of Cyberspace: Why Less May Be Better," in *2012 4th International Conference on Cyber Conflict*, pp. 1–13, 2012.

13. S. L. Pfleeger and R. K. Cunningham, "Why Measuring Security is Hard," *IEEE Security and Privacy*, vol. 8, no. 4, pp. 46–54, 2010.

14. M. R. Grimaila and L. W. Fortson, "Towards an Information Asset-Based Defensive Cyber Damage Assessment Process," in *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications, CISDA 2007*, pp. 206–212, 2007.

15. S. Jajodia, A. K. Ghosh, V. Subrahmanian, V. Swarup, C. Wang, and X. S. Wang, *Moving Target Defense II: Application of Game Theory and Adversarial Modeling*. Springer Publishing Company, Inc., 2014.

16. S. Phleeger, "Useful Cybersecurity Metrics," *IT Professional*, vol. 11, no. 3, pp. 38–45, 2009.

17. P. Xie, J. H. Li, X. Ou, P. Liu, and R. Levy, "Using Bayesian Networks for Cyber Security Analysis," in *2010 IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 211–220, 2010.

18. S. Noel, L. Wang, and A. Singhal, "Measuring Security Risk of Networks Using Attack Graphs," *International Journal of Next-Generation Computing*, vol. 1, no. 1, pp. 135–147, 2010.

19. M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. New York: Cambridge University Press, 2011.

20. Q. Wu, S. Shiva, S. Roy, C. Ellis, and V. Datla, "On Modeling and Simulation of Game Theory-Based Defense Mechanisms Against DoS and DDoS Attacks," *Proceedings of the 2010 Spring Simulation Multiconference on - SpringSim '10*, p. 10, 2010.

21. K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory*. Morgan & Claypool, 2008.

22. Y. Yuan, F. Sun, and H. Liu, "Resilient Control of Cyber-Physical Systems Against Intelligent Attacker: A Hierarchal Stackelberg Game Approach," *International Journal of Systems Science*, vol. 47, no. 9, pp. 2067–2077, 2016.

23. W. Jiang, Z. Tian, H. Zhang, and X. Song, "A Stochastic Game Theoretic Approach to Attack Prediction and Optimal Active Defense Strategy Decision," in *2008 IEEE International Conference on Networking, Sensing and Control*, pp. 648–653, April 2008.

24. Y. Ouyang, H. Tavafoghi, and D. Teneketzis, "Dynamic Games with Asymmetric Information: Common Information Based Perfect Bayesian Equilibria and Sequential Decomposition," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 222–237, 2017.

25. N. S. Kovach, A. S. Gibson, and G. B. Lamont, "Hypergame Theory: A Model for Conflict, Misperception, and Deception," *Game Theory*, 2015.

26. M. Connell and S. Vogler, "Russia's Approach to Cyber Warfare," *Centre for Naval Analysis Occasional Paper Series*, p. 32, March 2017.

27. S. M. Ross, *Stochastic Processes*. University of California, Berkeley: John Wiley & Sons, Inc., 2nd ed., 1996.

28. S. M. Ross, *Introduction to Probability Models*. University of Southern California: Elsevier, 11th ed., 2014.

29. M. Rouse, "Patch Tuesday." https://searchsecurity.techtarget.com/definition/Patch-Tuesday. July 2017. Retrieved on 10 July 2018.

30. Microsoft, "Understanding How to Use the Microsoft Security Response Center Exploitability Index." https://technet.microsoft.com/en-us/library/dd145265.aspx. Retrieved on 21 August 2018.

31. Microsoft, "Security Bulletins 2017." https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/securitybulletins2017. Retrieved on 10 July 2018.

32. E. Skoudis and T. Liston, *Counter Hack Reloaded*. Prentice Hall, 2nd ed., 2006.

33. R. D. McKelvey, A. M. McLennan, and T. L. Turocy, "Gambit: Software Tools for Game Theory, Version 16.0.1." http://www.gambit-project.org, 2016.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| **1. REPORT DATE** *(DD-MM-YYYY)* | **2. REPORT TYPE** | **3. DATES COVERED** *(From - To)* |
|---|---|---|
| 21-03-2019 | Master's Thesis | Sept 2017 – Mar 2019 |

**4. TITLE AND SUBTITLE**

A Stochastic Game Theoretical Model for Cyber Security

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Larkin, Michael T, Capt, USAF

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Institute of Technology
Graduate School of Engineering and Management
(AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT-ENS-MS-19-M-133

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Deputy Director Developmental Test, Evaluation & Prototyping
1400 Defense Pentagon
Washington, DC 20301-1400 thomas.w.simms2.civ@mail.mil

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

**13. SUPPLEMENTARY NOTES**



**14. ABSTRACT**

The resiliency of systems integrated through cyber networks is of utmost importance due to the reliance on these systems for critical services such as industrial control systems, nuclear production, and military weapons systems. Current research in cyber resiliency remains largely limited to methodologies utilizing a singular technique that is predominantly theoretical with limited examples given. This research uses notional data in presenting a novel approach to cyber system analysis and network resource allocation by leveraging multiple techniques including game theory, stochastic processes, and mathematical programming. An operational network security problem consisting of 20 tactical normal form games provides an assessment of the resiliency of a cyber defender's network by leveraging the solutions of each tactical game to inform transitional probabilities of a discrete-time Markov chain over an attacker-defender state space. Furthermore, the Markov chain provides an assessment of the conditional path through the operational problem with an expected cost of damage to the defender network. The solutions of the tactical games and, in turn the operational problem, are utilized to determine the effects and risks of projected network improvement resource allocation decisions via an integer program. These results can be used to inform network analysts of the resiliency of their network while providing recommendations and requirements for improving their network resiliency posture against potential malicious external actors.

**15. SUBJECT TERMS**

Game Theory, Markov Chains, Mathematical Programming, Cyber Security

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | Dr. Darryl K. Ahner, AFIT/ENS |
| U | U | U | UU | 141 | **19b. TELEPHONE NUMBER** *(include area code)* (937)255-3636,x4708; darryl.ahner@afit.edu |